

UNIVERSIDADE FEDERAL DO PARANÁ

CARLOS ALEXANDRE GOUVEA DA SILVA

**MÉTODO PARA ATENUAR A DEGRADAÇÃO DA QUALIDADE EM  
VÍDEO STREAMING SOBRE IP UTILIZANDO REDUNDÂNCIA  
SELETIVA E TRANSFERÊNCIA MULTICAMINHO**

CURITIBA

2015

CARLOS ALEXANDRE GOUVEA DA SILVA

**MÉTODO PARA ATENUAR A DEGRADAÇÃO DA QUALIDADE EM  
VÍDEO STREAMING SOBRE IP UTILIZANDO REDUNDÂNCIA  
SELETIVA E TRANSFERÊNCIA MULTICAMINHO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração Telecomunicações, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Carlos Marcelo Pedroso

Co-orientador: Prof. Dr. Eduardo Parente Ribeiro

CURITIBA

2015

---

S586m

Silva, Carlos Alexandre Gouvea da  
Método para atenuar a degradação da qualidade em vídeo streaming  
sobre IP utilizando redundância seletiva e transferência multicaminho/ Carlos  
Alexandre Gouvea da Silva. – Curitiba, 2015.  
85 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia,  
Programa de Pós-graduação em Engenharia Elétrica, 2015.

Orientador: Carlos Marcelo Pedroso – Co-orientador: Eduardo Parente  
Ribeiro.

Bibliografia: p. 80-85.

1. Transmissão digital. 2. Video digital - Qualidade. 3. Roteamento (Redes  
de computadores). I. Universidade Federal do Paraná. II. Pedroso, Carlos  
Marcelo. III. Ribeiro, Eduardo Parente. IV. Título.

CDD: 006.7

---

## TERMO DE APROVAÇÃO

CARLOS ALEXANDRE GOUVEA DA SILVA

MÉTODO PARA ATENUAR A DEGRADAÇÃO DA QUALIDADE EM  
VÍDEO STREAMING SOBRE IP UTILIZANDO REDUNDÂNCIA  
SELETIVA E TRANSFERÊNCIA MULTICAMINHO

Dissertação aprovada como requisito parcial à obtenção do grau de  
Mestre no Programa de Pós-Graduação em Engenharia Elétrica da Uni-  
versidade Federal do Paraná, pela seguinte banca examinadora:



Orientador: Prof. Dr. Carlos Marcelo Pedroso  
Departamento de Engenharia Elétrica, UFPR



Prof. Dr. Evelio Martín García Fernández  
Departamento de Engenharia Elétrica, UFPR



Prof. Dr. Luis Henrique Assumpção Lolis  
Departamento de Engenharia Elétrica, UFPR



Prof. Dr. Emilio Carlos Gomes Wille  
Departamento de Eletrônica, UTFPR

Curitiba, 10 de Julho de 2015

*Dedico este trabalho aquele que sempre me proporcionou todas as ferramentas e apoio para que eu pudesse seguir sempre em frente. Aquele que nos momentos de aflição e necessidade soube estender a mão sem hesitação. A essa incrível pessoa o meu obrigado eterno. Ao meu pai Sr. Luis Carlos o meu carinho e amor.*

## **AGRADECIMENTOS**

A Deus por me proporcionar saúde e forças para continuar almejando pelos meus objetivos.

A minha família pelo apoio nos momentos difíceis e turbulentos. Em especial a minha querida e amada mãe Célia por ser a base e grande incentivadora dos meus sonhos. Aos meus queridos e amados sobrinhos João Victor e Mariana Victória pela alegria de todos os dias.

Aos amigos e colegas, grandes incentivadores. Em especial ao grande amigo Edson Leonardo por partilhar dessa conquista em conjunto no programa de pós-graduação em Engenharia Elétrica.

Aos professores Carlos Pedroso e Eduardo Parente pelos ensinamentos levados para toda vida e os engenheiros eletricitas Amanda Biancchi Eloy e Alexandre Fonçatti pela grande ajuda no desenvolvimento e simulações realizadas.

## RESUMO

A transmissão de vídeo sobre IP é uma das aplicações mais importantes na Internet, com uma perspectiva de grande crescimento nos próximos anos. No entanto, em muitos casos, o controle sobre a qualidade de serviço nas redes de acesso é limitada, o que tende a reduzir a qualidade de reprodução de vídeo. Em paralelo, os sistemas de comunicação multiabrigados permitem o uso de diversas redes disponíveis, com a solução implementada na camada de transporte. O *Stream Control Transport Protocol* (SCTP) é um protocolo de camada de transporte capaz de utilizar múltiplas interfaces disponíveis. Nesta dissertação é proposto um método de transmissão de vídeo utilizando sistemas multiabrigados para transmitir pacotes redundantes, de modo a reduzir o impacto do aumento do atraso fim-a-fim na qualidade de reprodução do vídeo. O método foi desenvolvido para uso com vídeo codificado com padrão MPEG. A eficiência do método é demonstrada a partir de simulações computacionais, utilizando vídeos codificados com H.264. Os resultados indicam que o método proposto reduz significativamente a degradação da qualidade de reprodução do vídeo com custo pequeno em termos de envio de informações redundantes.

Palavras-chave: Vídeo Streaming, Qualidade percebida, Transmissão redundante.

## ABSTRACT

The video over IP is one of the most important applications on the Internet, with a growth trend in the coming years. However, in many cases, the control of quality of service in access networks is limited, which may impair the viewer's perceived quality. The multihomed systems allow the use of several available networks, with the solution implemented at the transport layer. The *Stream Control Transport Protocol* (SCTP) is a transport protocol able to use multiple available interfaces. We propose a method for video transmission over SCTP using redundant packets to reduce the impact of increase of end-to-end delay in viewer's perceived quality. The method was developed for use with video encoded with MPEG standard. The efficiency of the method is demonstrated from computer simulations using videos encoded with H.264. The results indicate that the proposed method significantly reduces the degradation of reproduction quality of the video with little cost in terms of transmission of redundant information.

Key words: Video Streaming, Perceived Quality, Transmission Redundant.



## LISTA DE FIGURAS

2.1	Sequência de formação do GoP. . . . .	21
2.2	Representação modelo de rede multiabrigada (HUANG; LIN, 2011). . .	24
2.3	Diagrama de sistema de avaliação e desempenho métrica SSIM (WANG et al., 2004). . . . .	26
3.1	Métodos de redundância de vídeo . . . . .	34
3.2	Função de autocorrelação (ACF) atrasos rede 3G, ADSL e WIFI. . . .	36
3.3	Ângulo $\theta$ de tendência de atraso. . . . .	37
3.4	Máquina de cinco estados para o método preditivo. . . . .	37
3.5	Máquina de dois estados para o método preditivo. . . . .	39
3.6	Algoritmo de seleção de quantidade de informação de cada tipo de quadro a ser enviada em cada GoP. . . . .	40
4.1	Esquemático de apresentação das fases de pré-processamento, transmissão e avaliação de vídeos simulados e testados. . . . .	42
4.2	Vídeo QCIF ( <i>Quarter Common Intermediate Format</i> ) utilizado: (A) Highway. 43	
4.3	Vídeos CIF ( <i>Common Intermediate Format</i> ) utilizados: (A) Akiyo, (B) Coastguard, (C) Football e (D) Bridge Close. . . . .	43
4.4	Vídeos HD utilizados: (A) Big Buck Bunny, (B) Crown Run e (C) Blue Sky. 43	
4.5	Modelo de rede utilizado pela recomendação G.1050. . . . .	45

## LISTA DE TABELAS

2.1	Comparação dos protocolos de transporte (BOKOR; HUSZAK; JENEY, 2009). . . . .	22
2.2	Mapeamento entre valores do PSNR e SSIM (ZINNER et al., 2010). . .	29
2.3	Comparação entre os trabalhos da literatura atual e o projeto de pesquisa.	32
3.1	Constantes ( $\rho$ ) de multiplicação de percentual de redundância da máquina de estados. . . . .	38
4.1	Características principais dos vídeos SD e HD utilizados para simulação e testes. . . . .	44
5.1	Níveis máximos de PSNR e SSIM possíveis para vídeos testados. . . .	49
5.2	Qualidade obtida a partir da transmissão 100% de redundância de vídeo no caminho secundário em 40 simulações e intervalo de confiança <i>t-student</i> 90%. . . . .	50
5.3	Resultados obtidos para $\alpha = 0,9$ , $\alpha = 1/2$ , $\alpha = 1/4$ , $\alpha = 1/6$ e $\alpha = 1/8$ combinados aos métodos relativos Linear, Exponencial e Quadrático para vídeo Akiyo. . . . .	51
5.4	Médias obtidas de PSNR e SSIM para transmissão redundante relativa para vídeos SD com $\alpha = 0,9$ . . . . .	53
5.5	Médias obtidas de PSNR e SSIM para transmissão redundante relativa para vídeos HD com $\alpha = 0,9$ . . . . .	54
5.6	Configurações de rede usados para gerador de atrasos baseado na recomendação G.1050. . . . .	55
5.7	Resultados obtidos para modelo gerador de atrasos, com <i>rede de acesso A</i> e <i>B</i> variável, recomendação G.1050 com $\alpha = 0,9$ e 95% de confiança para vídeo Highway. . . . .	56
5.8	Configurações dos parâmetros para comparação dos resultados com SCTP padrão. . . . .	58
5.9	Comparação resultados XU et al. SCTP padrão com método RSTM para o vídeo Highway QCIF. . . . .	58
5.10	Configurações dos parâmetros para comparação dos resultados com CMT e PR-CMT. . . . .	59
5.11	Comparação resultados XU et al. CMT e PR-CMT com método RSTM para o vídeo Highway QCIF. . . . .	59

5.12 Resultados obtidos para $\alpha = 0,9$ , $\alpha = 1/2$ , $\alpha = 1/4$ , $\alpha = 1/6$ e $\alpha = 1/8$ combinados à $\psi = 1$ , $\psi = 3$ , $\psi = 5$ e $\psi = 7$ para vídeo Coastguard. . . .	61
5.13 Resultados obtidos para $\alpha = 0,9$ , $\alpha = 1/2$ , $\alpha = 1/4$ , $\alpha = 1/6$ e $\alpha = 1/8$ combinados à máquina de 2 estados para vídeo Coastguard. . . . .	63
5.14 Resultados obtidos para $\alpha = 0,9$ , $\alpha = 1/2$ , $\alpha = 1/4$ , $\alpha = 1/6$ e $\alpha = 1/8$ combinados à $\psi = 1$ , $\psi = 3$ , $\psi = 5$ e $\psi = 7$ para vídeo Big Buck Bunny. .	64
5.15 Resultados obtidos para $\alpha = 0,9$ , $\alpha = 1/2$ , $\alpha = 1/4$ , $\alpha = 1/6$ e $\alpha = 1/8$ combinados à máquina de 2 estados para vídeo Big Buck Bunny. . . . .	65

## LISTA DE SIGLAS

3G	<i>Third Generation</i>
ACF	<i>Autocorrelation Function</i>
ACK	<i>Acknowledge</i>
ADSL	<i>Asymmetric Digital Subscriber Line</i>
ASU	<i>Arizona State University</i>
AVC	<i>Advanced Video Coding</i>
Cb	Crominancia azul
CIF	<i>Common Intermediate Format</i>
Cr	Crominancia vermelha
dB	Decibel
DCT	<i>Discrete Cosine Transform</i>
EMA	<i>Exponential Moving Averages</i>
FCC	<i>Federal Communications Commision</i>
Gbps	Giga bits por segundo
GOP	<i>Group of Pictures</i>
GVT	<i>Global Village Telecom</i>
HD	<i>High Definition</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
ICMP	<i>Internet Control Message Protocol</i>
IMS	<i>IP Multimidia Subsystem</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Television over IP</i>
ISO	<i>International Organization of Standardization</i>
ITU	<i>International Telecommunication Union</i>
LAN	<i>Local Area Network</i>

Mbps	Mega bits por segundo
MOS	<i>Mean Opinion Score</i>
MPEG	<i>Motion Picture Expert Group</i>
MSE	<i>Mean Squared Error</i>
MTU	<i>Maximum Transmission Unit</i>
PMR	<i>Path Maximum Retransmission</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
PR-SCTP	<i>Partial Reliability SCTP</i>
QCIF	<i>Quarter Common Intermediate Format</i>
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
RFC	<i>Request for Comments</i>
RSTM	Redundância Seletiva e Transferência Multicaminho
RTO	<i>Retransmission Timeout</i>
RTT	<i>Round Trip Time</i>
RTTVAR	<i>RTT Variance</i>
RTP	<i>Real-time Transport Protocol</i>
SCTP	<i>Stream Control Transport Protocol</i>
SD	<i>Standard Definition</i>
SLA	<i>Service-level Agreements</i>
SRTT	<i>Smoothed Round Trip Time</i>
SSIM	<i>Structural Similarity</i>
TCP	<i>Transport Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
UFPR	Universidade Federal do Parana
VCEG	<i>Video Coding Experts Group</i>
VoD	<i>Video on Demand</i>

## LISTA DE SÍMBOLOS

$\alpha$	Fator de amortecimento SRTT
$\delta$	Diferença atraso máximo e atraso suavizado
$\tau$	Atraso máximo permitido para atraso pacotes
$\lambda$	Parâmetro para modelo exponencial
$\theta$	Ângulo de inclinação da reta tangente ao SRTT em um dado ponto
$\psi$	Constante angular de troca de estado em máquina de estado
$\rho$	Fator de multiplicação da redundância
$\phi_G$	Quantidade média de informação GOP
$\phi_I$	Quantidade média de informação quadro I
$\phi_P$	Quantidade média de informação quadro P
$\phi_B$	Quantidade média de informação quadro B

## SUMÁRIO

<b>RESUMO</b>	<b>6</b>
<b>ABSTRACT</b>	<b>7</b>
<b>LISTA DE ILUSTRAÇÕES</b>	<b>8</b>
<b>LISTA DE TABELAS</b>	<b>10</b>
<b>1 INTRODUÇÃO</b>	<b>13</b>
1.1 Contexto . . . . .	13
1.2 Objetivos . . . . .	16
1.2.1 Objetivo geral . . . . .	16
1.2.2 Objetivos específicos . . . . .	16
1.3 Justificativa . . . . .	17
1.4 Estrutura da Dissertação . . . . .	18
<b>2 CONCEITOS BÁSICOS</b>	<b>19</b>
2.1 Padrão MPEG . . . . .	19
2.2 Protocolo SCTP . . . . .	21
2.3 Avaliação de Qualidade . . . . .	24
2.3.1 PSNR . . . . .	25
2.3.2 SSIM . . . . .	26

2.4	Trabalhos Relacionados . . . . .	29
<b>3</b>	<b>REDUNDÂNCIA SELETIVA E TRANSFERÊNCIA MULTICAMINHO</b>	<b>33</b>
3.1	Cálculo de Redundância . . . . .	33
3.1.1	Redundância Relativa . . . . .	34
3.1.2	Redundância Preditiva . . . . .	35
3.2	Seleção de Pacotes Redundantes . . . . .	39
<b>4</b>	<b>AVALIAÇÃO DE DESEMPENHO</b>	<b>41</b>
4.1	Materiais . . . . .	41
4.2	Metodologia . . . . .	41
4.2.1	Pré-processamento de imagem . . . . .	43
4.2.2	Transmissão e Simulação . . . . .	45
4.2.2.1	Transmissão . . . . .	46
4.2.3	Remontagem e Avaliação . . . . .	46
<b>5</b>	<b>ANÁLISE DOS RESULTADOS E DISCUSSÃO</b>	<b>48</b>
5.1	Transmissão <i>bicasting</i> de vídeo . . . . .	48
5.2	Transmissão em Redundância relativa . . . . .	50
5.2.1	Transmissão com tempos de atraso gerador G.1050 . . . . .	54
5.2.2	Comparação com trabalhos correlatos . . . . .	57
5.2.2.1	SCTP padrão . . . . .	57
5.2.2.2	CMT padrão e PR-CMT . . . . .	58
5.3	Transmissão redundância preditiva . . . . .	60



5.3.1	Transmissão preditiva (vídeo <i>Coastguard</i> ) . . . . .	60
5.3.2	Transmissão preditiva (vídeo <i>Big Buck Bunny</i> ) . . . . .	63
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>66</b>
<b>A</b>	<b>CÓDIGO FONTE DO GERADOR DE ATRASO REC G.1050</b>	<b>68</b>
<b>B</b>	<b>SCRIPT EM BASH</b>	<b>72</b>

## CAPÍTULO 1

### INTRODUÇÃO

#### 1.1 Contexto

Com a evolução das tecnologias de transmissão de dados e o aumento conjunto dos meios de acesso à Internet, houve aumento de oferta e procura de serviços de transmissão de vídeo. Dentre os serviços de transmissão amplamente utilizados, os sistemas de multimídia representam uma parcela significativa do uso dos ambientes de telecomunicações atuais. Dentre os sistemas de multimídia atuais pode-se mencionar o IPTV (*Internet Protocol Television*) e o VoD (*Video on Demand*).

O *video streaming* se refere à transmissão em tempo real de vídeo. Existem dois modos de transmissão de vídeo através da Internet, o modo de *download* e o modo de *streaming*. No modo de *download*, o usuário faz a transferência de todo o arquivo e em seguida reproduz o vídeo. No modo *download*, são eliminados os problemas decorrentes da variação do atraso e perdas na rede. Em contraste, no modo de *streaming* o vídeo é reproduzido apenas com um pequeno atraso. Exemplos de aplicações são IPTV, VoD, eventos esportivos, vídeo conferências ou telejornais. Devido à sua natureza em tempo real, o *streaming* de vídeo normalmente possui requisitos no que se refere a largura de banda, atraso e perda de pacotes. No entanto, a atual Internet de melhor esforço (*best effort*) não oferece qualquer garantia de qualidade de serviço (QoS-*Quality of Service*) para *streaming* de vídeo (WU et al., 2001).

Em contrapartida, o não atendimento dos requisitos de largura de banda exigidos pela transmissão de vídeo pode trazer inúmeros problemas para os usuários dos sistemas de multimídia. A degradação de parâmetros da rede como, o atraso, *jitter* e perda de pacotes, tem impacto direto na qualidade de vídeo percebida pelo usuário.

O atraso é o tempo absoluto que um pacote leva para ser transmitido de uma rede de origem a uma rede de destino (GREENGRASS; EVANS; BEGEN, 2009), o *jitter* é uma medida da variação do atraso (EPIPHANIOU et al., 2010) e a perda pacotes pode ser causada por problemas no meio físico ou por descarte de pacotes nos elementos de rede.

As arquiteturas de transmissão de multimídia, em sua grande maioria, devem prover níveis satisfatórios de QoS, definido como um conjunto de requisitos de serviços a serem cumpridas pela rede durante o transporte de uma conexão ou fluxo de dados (ITU-T, 2008a). Para (GREENGRASS; EVANS; BEGEN, 2009) os acordos de nível de serviço (SLA - *Service-level Agreements*) que asseguram a qualidade na transmissão de vídeo são definidos pelo atraso, jitter e perda de pacotes.

A qualidade de experiência (QoE, *Quality of Experience*) é definida em (ITU-T, 2007a) como o nível de qualidade de um aplicativo ou serviço, conforme percebida subjetivamente pelo usuário final. Existem diversos métodos objetivos disponíveis que tentam aproximar a avaliação subjetiva da QoE que podem ser implementadas sem a participação do usuário (SERRAL-GRACIÀ et al., 2010). A avaliação subjetiva é realizada a partir de avaliações humano-visuais e as objetivas são obtidas a partir de ferramentas computacionais.

Um dos fatores fundamentais que afetam a qualidade da transmissão de vídeo sobre a rede IP é o atraso fim-a-fim (FCC2014, 2014). O incremento neste atraso pode comprometer a qualidade dos serviços de voz, vídeo-conferência e vídeo. O *jitter* também é um dos fatores fundamentais que impactam na qualidade de tais sistemas (MAISONNEUVE et al., 2009) (CLAYPOOL; TANNER, 1999). De acordo com (DSL-FORUM, 2006), o atraso fim-a-fim e o *jitter* devem ser menores que 200ms e 50ms, respectivamente, para garantir uma boa qualidade de experiência na transmissão de vídeos codificados com MPEG4 (ISO-IEC, 1998) e MPEG2 (ISO-IEC, 1994).

Em função do aumento da diversidade dos meios de acesso à Internet surgiu a possibilidade de incorporar múltiplas interfaces de rede em um único dispositivo. Tais

sistemas são chamados multiabrigados. Um sistema é multiabrigado quando ele possui mais de uma interface de rede disponível para comunicação. Do ponto de vista da camada de transporte, cada interface representa um caminho diferente para que seja realizado o acesso à rede. Através de mecanismos de monitoramento e de troca de caminhos, é possível aumentar a resiliência e estabilidade de uma conexão, garantindo a continuidade dos serviços mesmo no caso do caminho principal se tornar indisponível. O SCTP (*Stream Control Transmission Protocol*) é um protocolo de camada de transporte que permite a utilização de múltiplos caminhos para transmissão de vídeo *streaming*.

Em contrapartida, os protocolos de camada de transporte mais utilizados atualmente são o *Transmission Control Protocol* (TCP) (DARPA, 1981) e o *User Datagram Protocol* (UDP) (POSTEL, 1980), sendo que o UDP é aplicado na transferência de vídeo e aplicações de tempo real sobre o *Real Time Transport Protocol* (RTP) (SCHULZRINNE; CASNER; FREDERICK, 2003). No entanto, os protocolos TCP e UDP não possuem capacidade de explorar a possibilidade de utilização de múltiplas interfaces.

O protocolo SCTP foi especificado pelo IETF (*Internet Engineering Task Force*) na RFC4960 (STEWART, 2007) e possui como principais características o suporte a sistemas multiabrigados (*multihoming*) e a transmissão por múltiplos fluxos (*multistreaming*). Por padrão, o SCTP utiliza um caminho para transmissão, chamado caminho primário, e deixa os demais em redundância. A RFC4960 especifica que os caminhos secundários podem ser utilizados para a retransmissão de pacotes, ou para *backup* em caso de falha do caminho principal. No entanto, os múltiplos caminhos também podem ser explorados para obter uma melhor qualidade para aplicações, que é o caso da proposta apresentada neste trabalho. Por padrão, o SCTP realiza a troca de caminho primário caso seja detectada a perda de 6 segmentos, sendo o segmento indicado como um pacote da camada de transporte. Um segmento é considerado perdido se a confirmação de recepção não for recebida até um certo tempo limite (RTO, *retrans-*

*mission time out*). Quando um segmento é declarado perdido, o RTO é dobrado. O valor mínimo inicial para o RTO é de 1 segundo. Desta forma, o tempo mínimo para troca de caminho primário é de 63 segundos, o que inviabiliza o uso do SCTP padrão para prevenir a degradação de qualidade de vídeo *streaming*.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

O objetivo geral é desenvolver um método de transmissão de vídeo *streaming* sobre IP utilizando sistemas de multicaminhos com envio de pacotes redundantes pelo caminho secundário para reduzir a degradação da qualidade causada pelo aumento do atraso fim-a-fim no caminho primário.

### 1.2.2 Objetivos específicos

Os objetivos específicos são:

- Desenvolver modelo matemático para o cálculo da quantidade de informações redundantes a serem enviadas pelo caminho secundário.
- Desenvolver um método para seleção de pacotes a serem transmitidos de forma redundante.
- Comparar e avaliar a qualidade dos vídeos a partir do PSNR (Peak Signal-to-Noise Ratio) e SSIM (Structural Similarity), utilizando ferramentas estatísticas para avaliação de resultados.

### 1.3 Justificativa

Após pesquisas na literatura, observou-se a existência de trabalhos que utilizam de sistemas multicaminhos para transmissão de vídeo com o objetivo de reduzir os impactos que resultam na má qualidade de visualização, porém sem a utilização dos múltiplos caminhos para transmissão redundante de vídeo *streaming*. Nesta dissertação, é proposto um método para transmissão de vídeo com envio de pacotes redundantes utilizando caminhos secundários com o objetivo de reduzir a degradação na qualidade causada pelo aumento do atraso no caminho primário. O método proposto é chamado de Redundância Seletiva e Transferência Multicaminho (RSTM).

A crescente demanda de dispositivos móveis, vem expandindo cada vez mais o total de tráfego dedicado a vídeo em rede móveis, segundo estudo realizado pela Cisco (CISCO, 2015). A previsão indica que em 2019 a aplicação de vídeo *streaming* gerará cerca de 79% do total do tráfego de redes móveis. Em 2018 a cada segundo será transmitido o equivalente a um milhão de minutos em vídeos. O aumento da resolução dos vídeos transmitidos também é previsto, que em 2018 deve contar com 11% do total de vídeos com padrão Ultra HD (4K).

As redes de acesso normalmente não fornecem garantias de qualidade de serviço para transmissão de vídeo, sendo o atraso de propagação uma das principais causas da degradação da qualidade, uma vez que a taxa de transmissão parece estar crescendo para atender os requisitos de transmissão de vídeo (FCC2014, 2014). A flexibilização e otimização de protocolos de transporte já permitem o uso de sistemas com duas ou mais interfaces de rede disponíveis. Os sistemas atuais não fazem uso de múltiplas interfaces de rede disponíveis para prevenir os efeitos negativos da degradação dos parâmetros de rede, que consequentemente impactam na qualidade percebida de reprodução de vídeo.

## **1.4 Estrutura da Dissertação**

Além deste capítulo introdutório, esta dissertação está organizada da seguinte forma. No Capítulo 2 são apresentados os fundamentos sobre codificação de vídeo utilizando MPEG, o panorama de funcionamento do SCTP, as métricas de avaliação da qualidade de vídeo e os trabalhos relacionados. No Capítulo 3 é descrito o método proposto RSTM. O Capítulo 4 apresenta a avaliação de desempenho do método. No Capítulo 5 são apresentados os resultados obtidos e respectiva discussão. Finalmente, no Capítulo 6 são apresentadas as conclusões e trabalhos futuros.

## CAPÍTULO 2

### CONCEITOS BÁSICOS

#### 2.1 Padrão MPEG

O *Motion Picture Expert Group* (MPEG), criado em 1988 pela *International Organization of Standardization* (ISO), tem desenvolvido uma série de padrões de codificação de áudio e vídeo conhecidos como MPEG. Os primeiros padrões de codificação de vídeo foram o MPEG-1 e MPEG-2, sendo estes os grandes precursores dos padrões de codificação atuais como MPEG-4, MPEG-7 e MPEG-21. O MPEG-1 utiliza de altas taxas de amostragem visando a qualidade, enquanto o MPEG-2 já permitia taxas de amostragem mais baixas (NOLL, 1997).

O grande avanço no uso dos padrões de codificação MPEG é a possibilidade de compactação do vídeo. Um quadro de uma imagem digital no padrão *Standard Definition* (SD) de resolução 640x480 pixels que usa 8 bits para representação de cada pixel em escala de cinza, se transmitido a uma frequência de 30 quadros por segundo (fps - *frames per second*) resultaria no uso de uma largura de banda de aproximadamente 73 Mbps. Tipicamente, vídeos sem compressão utilizam aproximadamente 270 Mbps para SD e 1.485 Gbps para *high-definition* (HD) de taxa de transmissão (GREENGRASS; EVANS; BEGEN, 2009).

O MPEG utiliza três técnicas fundamentais de compressão de vídeo: a subamostragem, a compressão espacial e a compressão temporal. A compressão de subamostragem reduz informações de cores não perceptíveis ao olho humano. A redundância espacial geralmente denominada compactação *intraquadros*, define-se como semelhanças em um mesmo quadro. A compactação espacial é realizada através da transformada do cosseno (DCT - *Discrete Cosine Transform*) (SIKORA, 1997). A redundância tem-



poral define-se como semelhança entre quadros. A compactação temporal ou também denominada compactação *interquadros* identifica as diferenças entre os quadros de um GoP (*Group of Pictures*) e possibilita que os quadros sejam descritos com base nas diferenças dos quadros anteriores.

O MPEG-4 parte 10 ou H.264 (ITU-T, 2014) é um padrão de codificação de vídeo, resultado de um projeto de cooperação entre a VCEG (*Video Coding Experts Group*) da ITU (*International Telecommunication Union*) e o MPEG. Um dos grandes propósitos do desenvolvimento do padrão H.264 foi solucionar vários problemas de codificação de vídeo dos padrões anteriores como o MPEG-2 e o H.263, entre estas fornecer uma boa qualidade no vídeo com metade da taxa de transmissão em relação aos padrões anteriores (WIEGAND et al., 2003). O H.264 explora redundâncias espaciais e temporais para compactar o vídeo. Utilizando o padrão H.264 para compactação de vídeo em *standard definition* (SD) é necessária uma taxa média de 1,75 Mbps e para vídeos em *high definition* (HD), 8 Mbps. Em comparação, o MPEG-2 utiliza aproximadamente 3,75 Mbps para SD e até 15 Mbps para HD (DSL-FORUM, 2006).

Para explorar a redundância temporal, o MPEG utiliza uma sequência pré definida de tipos diferentes de quadros chamada de grupo de figuras (GoP). Um GoP é formado por três tipos de quadros distintos. O quadro I (*intra*) é codificado sem a referência de outros quadros e usa apenas compressão espacial. O quadro P (*predictive*) é codificado a partir de um quadro I ou quadro P anterior. O quadro B (*bi-directionally*) utiliza informações do quadro P ou I anterior e do próximo quadro I ou P como referência para sua codificação.

O GoP tem início com um quadro I, que é seguido de diversos quadros B e P, como ilustrado na Figura 2.1. Para representação da estrutura de composição de quadros, é comum utilizar a notação  $(M, N)$ , onde M representa a quantidade de quadros por GoP e N representa a quantidade de quadros B consecutivos. Por exemplo  $(12, 2)$ , denota uma estrutura para um grupo de figuras formado pela sequência de quadros *IBBPBBPBBPBB*, que será repetida continuamente ao longo da duração do vídeo.

Como o quadro I é utilizado como referência para geração dos demais quadros do mesmo GoP, o seu nível de importância para decodificação é maior que os demais quadros.

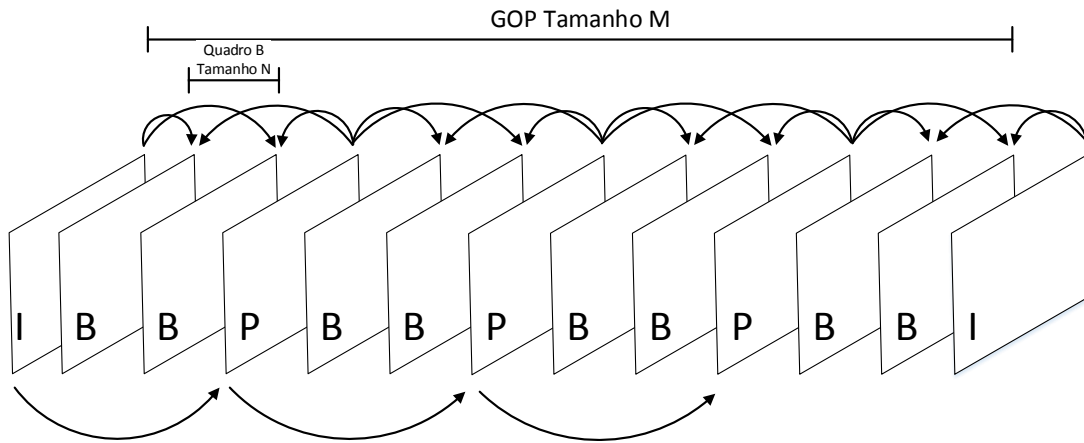


Figura 2.1: Sequência de formação do GoP.

## 2.2 Protocolo SCTP

O protocolo SCTP foi projetado para transportar mensagens de sinalização de telefonia sobre redes IP (STEWART, 2007), mas seu uso pode ser estendido a uma gama maior de aplicações. O SCTP é um protocolo de camada de transporte capaz de realizar a transmissão confiável ou não confiável, oferecer suporte à múltiplos *streams* em uma única conexão e também permite o uso de múltiplas interfaces de rede para aumentar a resiliência da conexão. A Tabela 2.1 apresenta a comparação entre os protocolos de camada de transporte TCP, UDP e SCTP.

O mecanismo de troca de caminho padrão utiliza apenas a perda de pacotes como critério para mudança de caminho. O protocolo utiliza um dos caminhos disponíveis para transmissão de dados, chamado de caminho primário. Uma mensagem de pulsação (ou *heartbeat*) é enviada periodicamente em todos os caminhos secundários. Um caminho é marcado como ativo depois que existe uma resposta do endereço requisitado. Caso não haja resposta de um determinado endereço este caminho é marcado como inativo. As mensagens de *heartbeat* podem ser enviadas em

Tabela 2.1: Comparação dos protocolos de transporte (BOKOR; HUSZAK; JENEY, 2009).

Característica	SCTP	TCP	UDP
<i>Orientado a conexão</i>	Sim	Sim	Não
<i>Confiável</i>	Sim	Sim	Não
<i>Entrega ordenada</i>	Sim/Não	Sim	Não
<i>Checksum</i>	Sim	Sim	Sim
<i>Tamanho checksum (bits)</i>	32	16	16
<i>Checksum parcial</i>	Não	Não	Não
<i>MTU no canal</i>	Sim	Sim	Não
<i>Controle de congestionamento</i>	Sim	Sim	Não
<i>Controle de fluxo</i>	Sim	Sim	Não
<i>Multihoming</i>	Sim	Não	Não
<i>Multistreaming</i>	Sim	Não	Não

intervalo de tempo configurável, sendo 30 segundos a configuração padrão do protocolo. Neste trabalho, é usando um *heartbeat* de 1 segundo de forma a obter os atrasos da rede posteriormente usados no cálculo de redundância dos vídeos.

Por padrão, o protocolo SCTP realiza a troca do caminho primário quando o número de pacotes descartados no mesmo caminho primário ultrapassar um certo limite. A troca de caminho como forma de melhorar os níveis de qualidade percebida pelo usuário não é alvo de investigação deste trabalho, mas o método RSTM pode ser empregado em conjunto com os algoritmos de troca de caminho.

O SCTP detecta perdas de pacotes monitorando o atraso de ida e volta (RTT, *Round Trip Time*), que é obtido pela diferença de tempo entre a transmissão da mensagem e o recebimento da confirmação de chegada no caminho primário e pelas confirmações das mensagens de *heartbeat* nos caminhos secundários. Para evitar variações bruscas do RTT, é calculado o SRTT (*Smoothed Round Trip Time*), utilizando uma média móvel exponencial (EMA, *Exponential Moving Averages*) definida a seguir:

$$\text{SRTT}_i = (1 - \alpha) \cdot \text{SRTT}_{i-1} + \alpha \cdot \text{RTT}_i \quad (2.1)$$

onde tipicamente  $\alpha = 1/8$  (STEWART, 2007).

A variância do RTT, chamada RTTVAR, é calculada da seguinte forma:

$$\text{RTTVAR}_i = (1 - \beta) \cdot \text{RTTVAR}_{i-1} + \beta \cdot |\text{SRTT}_i - \text{RTT}_i| \quad (2.2)$$

onde tipicamente  $\beta = 1/4$  (STEWART, 2007).

O protocolo SCTP detecta perdas utilizando um temporizador, chamado RTO (*Retransmission Timeout*). Caso a confirmação da transmissão da mensagem não seja recebida dentro deste tempo limite, ela será considerada perdida. Por padrão,  $\text{RTO} = \text{SRTT} + 4 \cdot (\text{RTTVAR})$  com valor mínimo definido em 1 segundo (STEWART, 2007).

Quando uma perda é detectada, um contador é incrementado. A variável PMR (*Path Maximum Retransmission*) representa o número máximo de perdas sucessivas do caminho primário toleradas pelo SCTP. Caso o número de perdas ultrapasse o PMR, um novo caminho será escolhido como primário. O SCTP utiliza como padrão  $\text{PMR}=5$ , e seu valor inicial é 0. Para concluir sobre a inatividade do caminho primário é necessário um tempo da ordem de 63 a 360 segundos. Isto é inviável para aplicações de vídeo e interatividade (WALLACE; SHAMI, 2012).

O SCTP propõe melhorias e soluções práticas com comparação aos protocolos de camada de transporte tradicionais como TCP e UDP. Um dos grandes avanços na concepção deste novo protocolo é a implementação do *multistreaming* e o *multihoming* (multicaminho) sendo descrito a seguir. Inúmeros trabalhos indicam que o uso do SCTP resulta na melhoria na qualidade de serviço em relação ao atraso e perda de pacotes (AFZAL et al., 2007).

Um dos grandes aprimoramentos do SCTP sobre os protocolos de camada de transporte tradicionais é capacidade multicaminho. Um *host* é dito multiabrigado caso exista mais de uma interface, física ou lógica, disponível no elemento de rede. Esta característica permite que uma associação SCTP possa ser mapeada em vários endereços IP (BOKOR; HUSZAK; JENEY, 2009). Um dos possíveis endereços IP é selecionado como primário, que é responsável em conduzir o tráfego da rede. A Figura

2.2 representa um modelo SCTP com múltiplos caminhos.

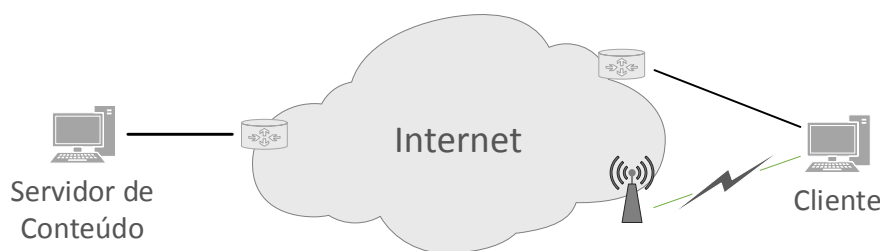


Figura 2.2: Representação modelo de rede multiabrigada (HUANG; LIN, 2011).

## 2.3 Avaliação de Qualidade

A qualidade pode ser descrita como a avaliação de um conjunto de características que diferenciam uma imagem original de uma imagem alterada. A diferença entre imagens ocorre devido ao fato que os sinais de vídeo estão sujeitos a degradação na sua captação, processamento, armazenamento e transporte.

As técnicas de avaliação da qualidade de vídeo são classificadas em subjetiva e objetiva. A avaliação subjetiva utiliza de observadores humanos para atribuir notas de avaliação ao vídeo assistido, que são posteriormente utilizadas para efetuar um cálculo estatístico cujo resultado é chamado de MOS (*Mean Option Score*). Contudo, essa avaliação utilizando-se de observadores se torna demorada e com alto custo. Esses custos são decorrentes de altos níveis de exigências dos ambientes criados para realização do testes, padronizados em (ITU-T, 2007c) (ITU-T, 2012) (ITU-T, 2008b). A avaliação objetiva utiliza de modelos matemáticos para estimar a qualidade do vídeo.

A avaliação objetiva de sinais de vídeo pode ser classificada em três categorias: completamente referenciada (FR - *Full Reference*), quando ambos os sinais, original e recebido, estão disponíveis para avaliação; parcialmente referenciada (RR - *Reduced Reference*), onde apenas algumas amostras do sinal original se encontra disponível; e não referenciada (NR - *No Reference*) quando somente o sinal de vídeo original está

disponível (FONSECA, 2008).

Nesta seção são descritos os métodos de avaliação objetiva utilizados na avaliação do método RSTM proposto nesta dissertação.

### 2.3.1 PSNR

O *Peak Signal-to-Noise Ratio* (PSNR) é um método objetivo para avaliação da qualidade que utiliza a relação entre o valor máximo possível de um sinal e o sinal de ruído distorcido que afeta a qualidade de sua representação entre os quadros de vídeo original e o de vídeo degradado. O PSNR é expresso em termos de decibéis (dB).

O cálculo do PSNR utiliza o erro médio quadrático (MSE - *Mean Squared Error*) obtido a partir da razão entre as somas do quadrado das diferenças de cada pixel do quadro de referência e do quadro degradado, e a quantidade de pixels do quadro, expresso por:

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - g(x,y)]^2 \quad (2.3)$$

onde  $M$  e  $N$  representam a largura e altura do quadro, respectivamente. A imagem original é representada por  $f(x,y)$  e a imagem degradada por  $g(x,y)$ . As coordenadas horizontal e vertical são denotadas por  $x$  e  $y$ , respectivamente. Quanto menor o valor do MSE melhor será a imagem degradada.

O PSNR é obtido através de:

$$PSNR = 10 \cdot \log_{10} \frac{L_{max}^2}{MSE} \quad (2.4)$$

onde  $L_{max} = 2^n - 1$  é o maior valor que pode ser assumido por um pixel e  $n$  é a quantidade de bits por pixel. Normalmente, em imagens representadas por oito bits ( $n = 8$ ) o valor de  $L_{max}$  é igual a 255.

O PSNR é representado em uma faixa de valores que varia entre zero e infinito, onde quanto maior o seu valor, maior a semelhança entre a imagem original e a degradada.

### 2.3.2 SSIM

O índice de similaridade estrutural (SSIM - *Structural Similarity*) é um método desenvolvido por (WANG; BOVIK, 2002) para estimar a qualidade da imagem a partir da combinação de três fatores: a perda de correlação, distorção de luminância e contraste. Esta métrica baseada na similaridade tem o objetivo de melhorar a correlação com as medidas subjetivas (WANG et al., 2004). O diagrama do sistema de avaliação da qualidade do SSIM é mostrado na Figura 2.3.

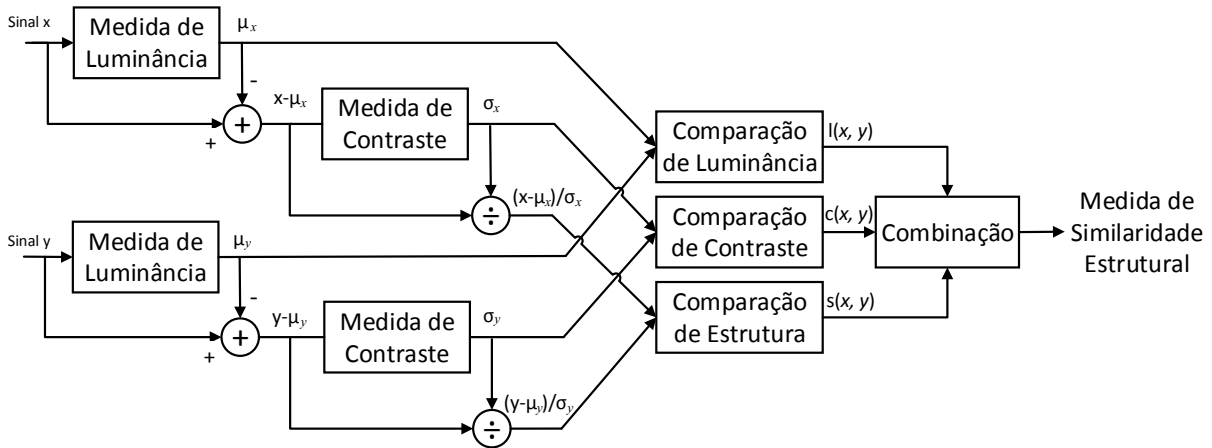


Figura 2.3: Diagrama de sistema de avaliação e desempenho métrica SSIM (WANG et al., 2004).

O SSIM realiza a medida de similaridade em três comparações: de luminância, de contraste e de estrutura. Primeiramente a luminância de cada sinal é comparada e calculadas as média  $\mu_x$  e  $\mu_y$ .

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.5)$$

$$\mu_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.6)$$

A seguir, são calculados  $x - \mu_x$  e  $y - \mu_y$  que correspondem à projeção do vetor para o hiperplano definido por:

$$\sum_{i=1}^N x_i = 0 \quad (2.7)$$

$$\sum_{i=1}^N y_i = 0 \quad (2.8)$$

O desvio padrão é usado como uma estimativa do sinal de contraste, estimado por:

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2} \quad (2.9)$$

$$\sigma_y = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2} \quad (2.10)$$

A comparação de contraste da imagem  $c(x,y)$  é então a comparação de  $\sigma_x$  e  $\sigma_y$ .

O sinal é normalizado pelo seu próprio desvio padrão, de modo que os dois sinais a serem comparados possuam desvio padrão unitário. A comparação da estrutura  $s(x,y)$  é realizado nos sinais normalizados  $(x - \mu_x)/\sigma_x$  e  $(y - \mu_y)/\sigma_y$ .

Finalmente, os três componentes são combinados para se obter uma medida geral de similaridade, representado por:

$$S(x,y) = f(l(x,y), c(x,y), s(x,y)) \quad (2.11)$$

As medidas de luminância, contraste e estrutura são definidas respectivamente



por:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (2.12)$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (2.13)$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (2.14)$$

onde as constantes  $C_1$ ,  $C_2$  e  $C_3$  são inseridas para evitar a instabilidade quando  $\mu_x^2 + \mu_y^2$ ,  $\sigma_x^2 + \sigma_y^2$  e  $\sigma_x\sigma_y$  estão muito próximos de zero . Especificamente, para essas constantes são usados (WANG et al., 2004):

$$C_1 = (K_1L)^2 \quad (2.15)$$

$$C_2 = (K_2L)^2 \quad (2.16)$$

$$C_3 = \frac{C_2}{2} \quad (2.17)$$

onde  $L$  é a faixa dinâmica dos valores de pixels (255 para imagens em tons de cinza de 8 *bits*) , e  $K_1 = K_2 \ll 1$  é uma pequena constante. Como no dimensionamento da luminância e contraste são introduzidas uma constante no denominador e numerador, no dimensionamento da estrutura é inserido na forma discreta  $\sigma_{xy}$  definida por:

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (2.18)$$

Combinando as três comparações 2.12, 2.13 e 2.14 é possível obter o índice de

similaridade entre os sinais  $x$  e  $y$  em função da luminância, contraste e estrutura:

$$SSIM(x,y) = [l(x,y)]^\epsilon \cdot [c(x,y)]^\omega \cdot [s(x,y)]^\gamma \quad (2.19)$$

onde  $\epsilon > 0$ ,  $\omega > 0$  e  $\gamma > 0$  são parâmetros usados para ajustar a importância relativa dos três componentes. O resultado do índice SSIM final também pode ser expresso detalhadamente em função das médias e dos desvios padrão, como segue.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.20)$$

A métrica do SSIM varia numa faixa de valores entre zero e um, com o valor 1 indicando o maior grau de semelhança e 0 indicando o menor.

A Tabela 2.2 apresenta a comparação entre os diferentes níveis de avaliação de qualidade objetiva a partir dos valores do PSNR e SSIM.

Tabela 2.2: Mapeamento entre valores do PSNR e SSIM (ZINNER et al., 2010).

PSNR (dB)	SSIM	Significado
$PSNR \geq 45$	$SSIM \geq 0,99$	Excelente
$33 \leq PSNR < 45$	$0,95 \leq SSIM < 0,99$	Bom
$27,4 \leq PSNR < 33$	$0,88 \leq SSIM < 0,95$	Razoável
$18,7 \leq PSNR < 27,4$	$0,5 \leq SSIM < 0,88$	Pobre
$18,7 < PSNR$	$0,5 < SSIM$	Ruim

## 2.4 Trabalhos Relacionados

Atualmente a literatura científica apresenta um avanço significativo nos trabalhos envolvendo o uso de sistemas de multicaminhos na transmissão de conteúdo multimídia como vídeo *streaming*.

Em (TORRES, 2014) é proposto um método para melhoria da qualidade percebida na transmissão de vídeo *streaming* a partir da seleção de caminho primário para sistemas multiabrigados com o protocolo SCTP. Neste método é proposto o *Predic-*

*tive Delay-Centric* PDC, uma adaptação do *Delay-Centric* (KELLY et al., 2004), para previsão da tendência no aumento de latência e assim melhorar a estabilidade e desempenho em sistemas de vídeo *streaming*. A melhoria de QoE também é estudada por (SANTOS, 2012) através de um método que realiza o reconhecimento da carga útil dos pacotes de vídeo em fluxos individuais de vídeo e, em caso de congestionamento, realiza o descarte seletivo de pacotes não prioritários. O reconhecimento foi realizado com o auxílio de redes neurais devido à baixa complexidade computacional. Em ambos os métodos houve a melhora de QoE.

Em (MOLTENI; VILLARI, 2002) são exploradas características do SCTP com *Partial Reliability* (PR-SCTP) (STEWART et al., 2004) para retransmissão de dados prioritários utilizando vídeo codificado com MPEG4. A proposta altera as políticas de retransmissões, estabelecendo um tempo limite para realizar tentativas de retransmissão, reduzindo o congestionamento gerado por tentativas de retransmissão de dados que já não serão mais úteis para reprodução do vídeo. Os resultados apresentados indicaram uma melhora do PSNR se comparado ao uso do protocolo UDP. Esta técnica não tira proveito dos múltiplos caminhos disponíveis para a transmissão.

O PR-SCTP também é utilizado por (BOUSSEN et al., 2011) na adaptação do tráfego de IPTV sobre redes IMS (*IP Multimedia Subsystem*) com aprimoramento do processo de adaptação de parâmetros de qualidade de serviço, de forma a formatar o fluxo de vídeo de acordo com o SLA. Os resultados indicam melhora significativa dos padrões de qualidade percebida pelo usuário.

O PR-SCTP é usado também por (SANSON et al., 2010) onde é proposto um modelo probabilístico para antecipar possíveis retransmissões utilizando caminhos secundários, prevendo assim perdas de pacotes, e reduzindo as consequências da perda pelo caminho primário. Um comparativo entre os diferentes protocolos de transporte são avaliados em (KIM; KOH; KIM, 2007) na aplicação de sistemas IPTV, onde o SCTP se mostrou mais adequado em se tratando de conteúdo com transmissão em tempo real.

Os autores (HUANG; LIN, 2011) propõem uma transferência multicaminho parcialmente confiável concorrente PR-CMT (*Concurrent Multipath Transfer*). O PR-CMT combina técnicas de transferência simultânea de múltiplos caminhos do CMT e a transmissão parcialmente confiável da PR-SCTP em fluxos priorizados. PR-CMT permite transmitir quadros importantes antes de outros quadros, mas não transmite quadros com tempo de vida expirados. Os resultados das simulações mostram que o PR-CMT pode inferir precisamente na transmissão de dados prioritários e mais importante ainda, pode evitar grandes lacunas entre dois quadros transmitidos para obter boa qualidade de vídeo .

Em (XU et al., 2011) os autores avaliam a utilização do CMT que permite a transmissão simultânea por outros caminhos na rede e o PR-CMT (*Partial Reliability*) que além de permitir a transmissão simultânea de vídeo *streaming* é aplicado a uma política de confiabilidade parcial de retransmissão de dados prioritárias. Os resultados indicaram que a utilização do PR-CMT apresentou melhores níveis de PSNR do que o CMT padrão.

Em (TARABUTA; POTORAC; BALAN, 2013) é proposto um método de transmissão com redundância total utilizando múltiplos caminhos simultaneamente, usando infraestrutura de rede de Internet. O método proposto utiliza pelo menos o dobro da largura de banda para envio dos dados. Neste trabalho não foi apresentada avaliação de desempenho em termos de qualidade de experiência. No entanto, apresenta dados sobre a melhoria de parâmetros de qualidade de serviço, como perda de pacotes e atraso.

Okamoto et al. (OKAMOTO et al., 2014) propõem a transmissão duplicada (*bicasting*) de segmentos SCTP que contenham informação de intervalos de confirmação (*ACK gaps*). Assim, a perda destes pacotes em apenas um dos caminhos, previne problemas com atraso devido à espera por confirmações, o que levaria a uma redução na qualidade de reprodução. Eles demonstraram as vantagens desta estratégia utilizando um experimento com enlaces sem fio. Entretanto, esta abordagem não prevê a

transmissão duplicada de outros segmentos que possam conter dados importantes.

Neste trabalho, pretende-se aplicar o método RSTM de transmissão simultânea de pacotes redundantes. Esta transmissão usa características de multicaminho do protocolo SCTP com o objetivo de mitigar a degradação da qualidade de vídeo causada pelo atraso fim-a-fim do caminho primário. Os resultados obtidos deste trabalho são comparados aos resultados dos trabalhos encontrados na literatura atual.

A Tabela 2.3 apresenta uma comparação entre as principais características dos trabalhos relacionados ao método proposto neste trabalho. Dentre os trabalhos que serão usados na comparação de métodos está o (XU et al., 2011) que apresenta resultados com PSNR e usa múltiplos caminhos porém não redundantes.

Tabela 2.3: Comparação entre os trabalhos da literatura atual e o projeto de pesquisa.

Proposta	Codec	Multicaminho	Transmissão redundante	Avaliação qualidade	Método
TORRES	H.246	Sim	Não	PSNR / SSIM	PDC
MOLTENI; VIL-LARI	MPEG-4	Não	Não	PSNR	PR-SCTP
BOUSSEN et al.	Não informado	Sim	Não	Não	PR-SCTP
SANSON et al.	H.264	Sim	Não	Não	PR-SCTP
KIM; KOH; KIM	MPEG-TS	Não	Não	Não	SCTP
HUANG; LIN	MPEG-4	Sim	Não	PSNR	PR-CMT
XU et al.	MPEG-4	Sim	Não	PSNR	PR-CMT
TARABUTA; POTORAC; BALAN	Não informado	Sim	Sim	Não	Redundância total
OKAMOTO et al.	Não informado	Sim	Sim	Não	SCTP
RSTM	H.246	Sim	Sim	PSNR / SSIM	SRTM-SCTP

## CAPÍTULO 3

### REDUNDÂNCIA SELETIVA E TRANSFERÊNCIA MULTICAMINHO

O RSTM consiste na transmissão de pacotes redundantes utilizando os caminhos secundários, enquanto o mesmo vídeo é transmitido pelo caminho primário. Este trabalho propõe um método de cálculo da quantidade de informação redundante a ser transmitida, sendo obtida a partir do tempo de atraso fim-a-fim do caminho primário. O percentual de redundância permite selecionar um conjunto de pacotes de vídeo a ser enviado simultaneamente pelos caminhos secundários. São utilizados dois modelos de cálculo de redundância, baseado em funções de mapeamento e em um modelo preditivo de previsão de tendência de atraso.

#### 3.1 Cálculo de Redundância

O cálculo de redundância é realizado a partir dos tempos de atraso amostrados pelo *heartbeat*, nativo do SCTP, nos caminhos secundários ativos. O atraso fim-a-fim é estimado tomando-se a metade do RTT. O valor obtido de redundância define a quantidade de informação do vídeo que deve ser enviada pelo caminho secundário. Como o atraso pode ser variável na rede, e em alguns intervalos abruptamente, é feita a suavização do atraso na rede a partir do SRTT com indicado na equação 2.1. O nível de suavização é obtido a partir da constante  $\alpha$ , onde quanto mais próximo de 1 representa suavização fraca e mais próximo a 0 representa suavização forte. Neste trabalho foram realizados testes para suavização com  $\alpha$  igual a 1/8, 1/6, 1/4, 1/2 e 0,9.

A escolha da constante  $\alpha$  para suavização do RTT afeta o desempenho do método proposto. O  $\alpha$  próximo ao valor 1 resulta em uma suavização fraca, indicando que o

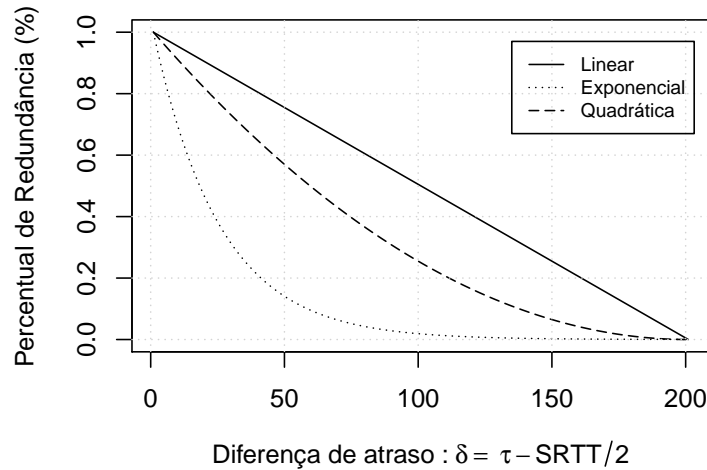


Figura 3.1: Métodos de redundância de vídeo

SRTT não filtra variações abruptas do RTT. A diminuição do valor de  $\alpha$  implica em uma suavização mais forte, permitindo eliminar as mudanças abruptas do atraso.

O SRTT é aplicado à dois diferentes métodos de cálculo de redundância. No primeiro método é considerado somente o valor atual do SRTT, aqui chamado de método de redundância relativa. O segundo método é baseado na predição de tendência do SRTT, chamado de método de redundância preditiva. Esses dois métodos são descritos a seguir.

### 3.1.1 Redundância Relativa

O percentual de informação redundante a ser transmitida no segundo seguinte é obtido através de uma função de mapeamento utilizando a diferença entre o valor do atraso fim-a-fim suavizado e um valor de tempo limite, dado por  $\delta = \tau - \text{SRTT}/2$ . O valor de  $\delta$  é utilizado para obter o percentual de informação redundante através de uma função de mapeamento. Neste trabalho, foram analisadas três funções de mapeamento: linear, quadrática e exponencial, conforme ilustrado no gráfico da Figura 3.1.

As funções de mapeamento linear, quadrática e exponencial são denotadas respectivamente pelas equações 3.1, 3.2 e 3.3. Estas funções são obtidas a partir do parâmetro  $\tau$  que indica o limite máximo aceitável de atraso da rede para vídeo *streaming*.

ming. As três funções em estudo são definidas por:

$$P_{linear}(\delta) = \begin{cases} \frac{\tau - \delta}{\tau} & \delta \geq 0 \\ 1 & \delta < 0 \end{cases} \quad (3.1)$$

$$P_{quad}(\delta) = \begin{cases} \left(\frac{\tau - \delta}{\tau}\right)^2 & \delta \geq 0 \\ 1 & \delta < 0 \end{cases} \quad (3.2)$$

$$P_{exp}(\delta) = \begin{cases} e^{-\lambda\delta} & \delta \geq 0 \\ 1 & \delta < 0 \end{cases} \quad (3.3)$$

onde P representa a quantidade de informação redundante a ser enviada no próximo segundo,  $\lambda = 1/25$  e  $\tau = 200ms$ .

### 3.1.2 Redundância Preditiva

Ao utilizar um modelo baseado na tendência do atraso, deve ser analisada a dependência temporal da série de atraso. A Figura 3.2 apresenta a função de autocorrelação (ACF) observada nos atrasos medidos nas três redes em estudo. Observa-se que em todos os casos existe uma dependência temporal significativa na série, o que indica que podem ser realizadas previsões com base em valores passados. Esta análise permite utilizar de modelos de predição para o cálculo da redundância, e que neste trabalho é descrito a partir da aplicação de uma máquina de estados.

A redundância preditiva é uma combinação do modelo relativo, descrito anteriormente, com a análise preditiva. Um Modelo Preditivo analisa um conjunto de dados atuais e passados para definir uma regra que permita predizer um determinado resultado futuro. Tipicamente os modelos de predição utilizam de fundamentos estatísticos como ferramenta principal.

O objetivo do uso do modelo preditivo ocorre a partir da necessidade de estimar a



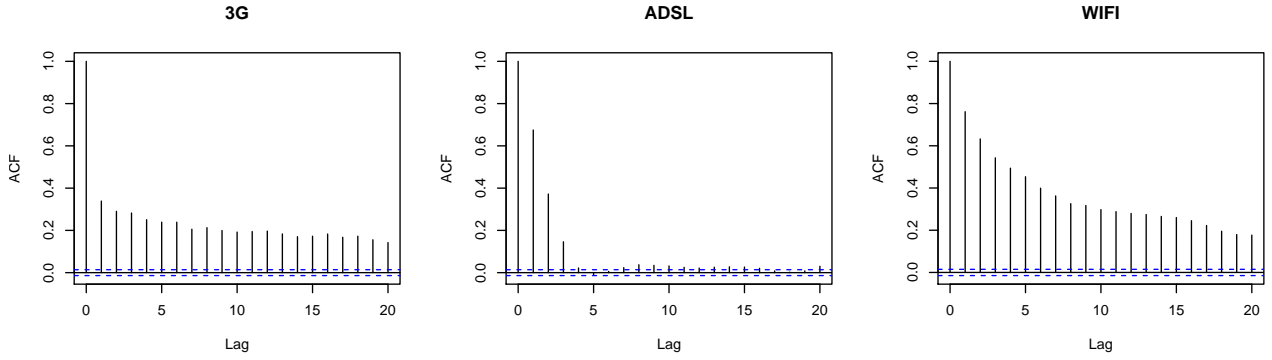


Figura 3.2: Função de autocorrelação (ACF) atrasos rede 3G, ADSL e WIFI.

redundância necessária decorrente da tendência de aumento ou diminuição do atraso fim-a-fim na rede do caminho primário. Assim, deseja-se reduzir a quantidade de informação redundante para a mínima possível e melhorar a qualidade da imagem quando o aumento do atraso puder ser previsto.

Com o uso da máquina de estados é considerado somente o atraso no tempo  $t$  e no tempo  $t-1$ , não consumindo recursos de hardware excessivos.

Neste trabalho, são definidas três possíveis tendências de atraso: a primeira quando o atraso aumenta, segundo quando o atraso diminui por fim quando o atraso se mantém constante. O coeficiente angular, mostrado em 3.4, é obtido a partir de  $y_2$  e  $y_1$  que representam o variação do atraso, respectivamente do tempo anterior  $x_1$  para o tempo  $x_2$ .

Sendo o atraso da rede variável no tempo, é possível que os atrasos aumentem ou diminuam rapidamente ou lentamente comparando o atraso amostrado anterior para o atual. Nesse panorama, é calculado o ângulo  $\theta$  da linha do tempo de atraso em relação ao intervalo de amostragem anterior, como visto na Figura 3.3. O ângulo  $\theta$  indica o sentido de tendência do atraso, onde o atraso aumenta quando  $\theta > 0$ , o diminui quando  $\theta < 0$  ou são iguais quando  $\theta = 0$ .

O ângulo  $\theta$  é obtido através de manipulações trigonométricas com:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3.4)$$

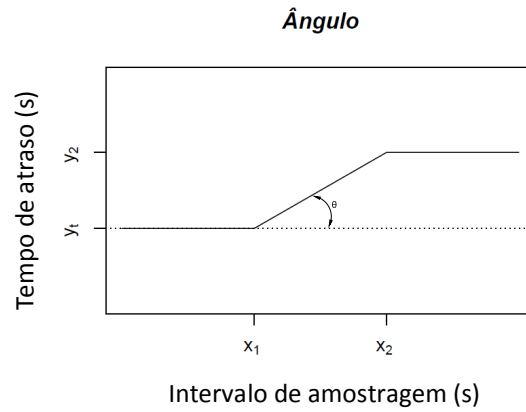


Figura 3.3: Ângulo  $\theta$  de tendência de atraso.

onde  $m$  é o coeficiente angular da reta. Como as amostras de atrasos foram coletadas em intervalos de 1 *segundo* tem-se:

$$m = \tan \theta \quad (3.5)$$

Logo, o ângulo  $\theta$  é obtido através de.

$$\theta = \arctan(m) \quad (3.6)$$

O ângulo  $\theta$  é aplicado em uma máquina de estados para definição de um multiplicador ao valor de redundância relativa calculado anteriormente. A máquina de estados é apresentada na Figura 3.4.

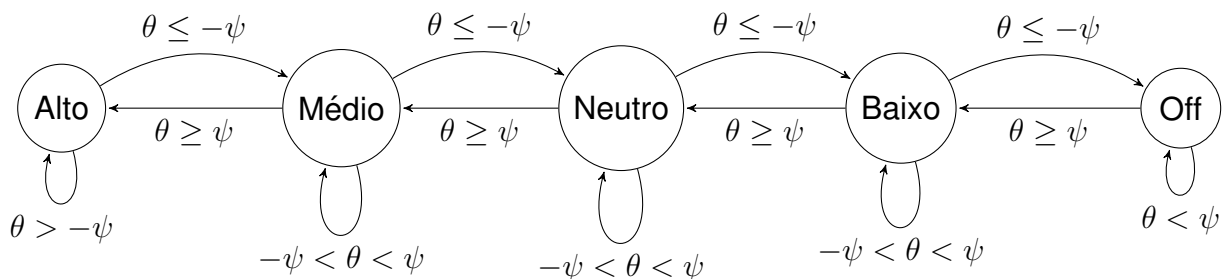


Figura 3.4: Máquina de cinco estados para o método preditivo.

É incluída uma constante  $\psi$  para evitar a transição de estados com variações não significativas. Na prática, quando usado um  $\psi$  baixo ocorrerá a troca de estado mais

facilmente. Para um  $\psi$  muito alto, a máquina de estados somente troca de estado quando ocorrer um aumento ou queda muito abrupto no atraso fim-a-fim. Neste estudo, foram avaliados valores de  $\psi$  iguais a 1, 3, 5 e 7. Na prática quanto maior o valor de  $\psi$ , menos transições ocorrerão na máquina de estados.

Cada estado na máquina anterior define uma constante de multiplicação pelo percentual de redundância calculado no modelo relativo. As constantes de multiplicação para cada estado são apresentadas na Tabela 3.1. O percentual de redundância final é dado pelo produto da constante de multiplicação  $\rho$  e o percentual de redundância relativo  $P_w(\delta)$ , sendo representado pela equação:

$$P_{final} = \rho P_w(\delta) \quad (3.7)$$

onde  $w$  indica a função de mapeamento usada ( *Linear*, *Exponencial* ou *Quadrática*).

Tabela 3.1: Constantes ( $\rho$ ) de multiplicação de percentual de redundância da máquina de estados.

Estado	Constante ( $\rho$ )
Alto	2
Médio	1.5
Neutro	1
Baixo	0.5
Off	0

A utilização do modelo preditivo também é aplicada a uma máquina de dois estados. Neste modelo, considera-se como critério de troca de estado a tendência de aumento ou queda do SRTT. O percentual de redundância é dado por  $P_w(\delta)$  para o estado On e zero para o estado Off. A diferença entre os atrasos suavizados é dada por  $dSRTT = SRTT_i - SRTT_{i-1}$ . A máquina de dois estados respectiva para este método é apresentado na Figura 3.5.

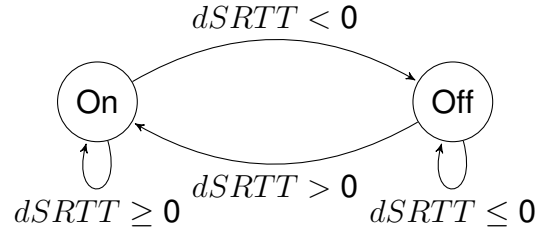


Figura 3.5: Máquina de dois estados para o método preditivo.

### 3.2 Seleção de Pacotes Redundantes

A seleção dos pacotes de vídeo redundantes inicia-se com o cálculo do percentual de informação redundante a ser enviado. No caminho primário são enviados todos os pacotes de vídeo. No caminho secundário, onde ocorre o envio redundante, os pacotes são enviados de acordo com o percentual de redundância calculado a partir do atraso fim-a-fim da rede primária. Como descrito no Capítulo 1 os quadros de vídeo possuem diferentes níveis de importância entre si, sendo os mais importantes os quadros *I*, seguidos pelos quadros *P* e *B*. Os quadros *P* mais importantes são os mais próximos do início do GoP.

A quantidade média de informação  $\phi_G$  em bytes de um GoP pode ser escrita como

$$\phi_G = \phi_I + W\phi_P + Z\phi_B \quad (3.8)$$

onde a quantidade média de informação em bytes no quadro *I*, *P* e *B* são dados respectivamente por  $\phi_I$ ,  $\phi_P$  e  $\phi_B$ . E o número de quadros *P* e *B* no GoP são denotados respectivamente por *W* e *Z*.

O algoritmo mostrado na Figura 3.6 apresenta o método proposto para seleção da informação redundante. Considera-se conhecido o valor de  $\phi_I$ ,  $\phi_P$  e  $\phi_B$ , que são as entradas do algoritmo. O valor de redundância  $P(\delta)$  também é entrada do algoritmo, que deve ser executado a cada segundo. As variáveis  $E_I$ ,  $E_P$ ,  $E_B$  são as saídas e representam respectivamente a quantidade de informação redundante de quadros *I*, *P* e *B* a serem enviadas em cada um dos GoPs do segundo seguinte, ou seja, pacotes de

quadros e não os quadros inteiros. O algoritmo apresentado possui baixa complexidade computacional, de ordem  $O(1)$ , e exige o conhecimento do tipo de quadro sendo transmitido.

---

```

1 Input:  $P(\delta)$ ; // percentual de redundância
2 Input:  $\phi_G, \phi_I, \phi_P, \phi_B$ ; // média de informação no GOP e em
   quadros I, P e B
3 Output:  $E_I, E_P, E_B$ ; // quantidade de informação de
   quadros I, P e B a serem enviados por GOP
4 if  $P(\delta) < \phi_I / \phi_G$  then
5    $E_I = \phi_G \cdot P(\delta)$ ;
6    $E_P = 0$ ;
7    $E_B = 0$ ;
8 else
9   if  $P(\delta) < (\phi_I + \phi_P) / \phi_G$  then
10     $E_I = \phi_I$ ;
11     $E_P = \phi_G \cdot P(\delta) - \phi_I$ ;
12     $E_B = 0$ ;
13  else
14     $E_I = \phi_I$ ;
15     $E_P = \phi_P$ ;
16     $E_B = \phi_G \cdot P(\delta) - \phi_I - \phi_P$ ;

```

---

Figura 3.6: Algoritmo de seleção de quantidade de informação de cada tipo de quadro a ser enviada em cada GoP.

## CAPÍTULO 4

### AVALIAÇÃO DE DESEMPENHO

Neste capítulo é apresentada a avaliação do desempenho do método RSTM. As diferentes ferramentas utilizadas são descritas.

#### 4.1 Materiais

A compilação dos dados e resultados apresentados nesta dissertação, foram resultados de diversas simulações e testes utilizando o Evalvid (KLAUE; RATHKE; WOLISZ, 2003). O Evalvid é um *framework* que contém um conjunto de ferramentas utilizadas para avaliação da qualidade de vídeo, e vem sendo utilizadas em diversas pesquisas relacionadas a vídeo *streaming* (LIE; KLAUE, 2008) (KLEIN; KLAUE, 2009) (JAMA et al., 2010) (TKN, 2014). As simulações foram realizadas em um servidor Linux, já com as ferramentas do Evalvid previamente instaladas. Os vídeos são codificados a partir do FFMPEG (COUDURIER et al., 2015), além de programas desenvolvidos em linguagem C para geração de atrasos e *scripts* de implementação e execução do método RSTM em *bash* disponível no Apêndice B.

#### 4.2 Metodologia

A metodologia utilizada será descrita em três etapas: a primeira descreve o pré-processamento do vídeo usado para cada simulação e teste; a segunda etapa apresenta o processo de transmissão do vídeo; e na última etapa são indicadas a remontagem do vídeo recebido e a avaliação da qualidade. A Figura 4.1 descreve a disposição das etapas de realização das simulações.

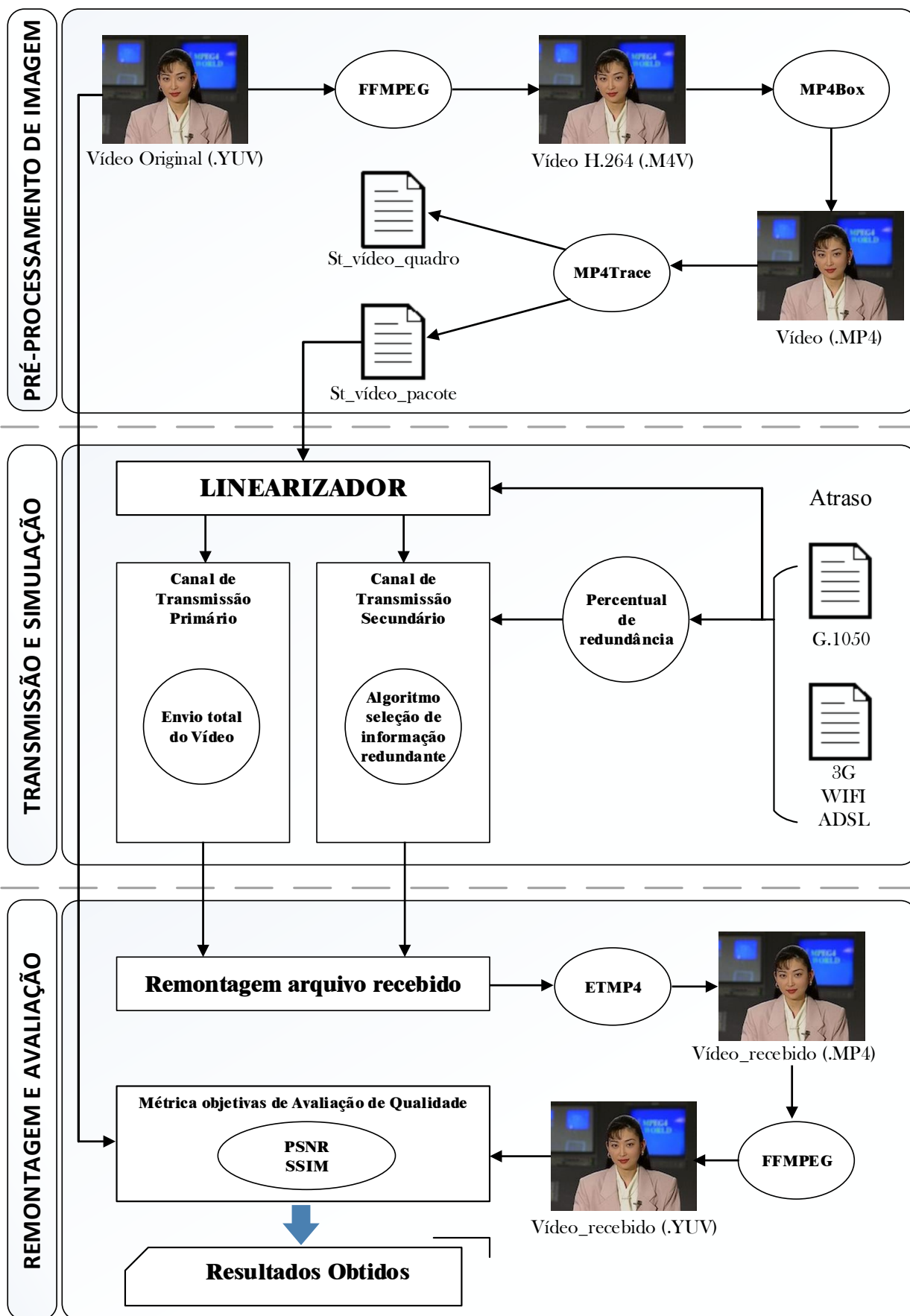


Figura 4.1: Esquemático de apresentação das fases de pré-processamento, transmissão e avaliação de vídeos simulados e testados.

### 4.2.1 Pré-processamento de imagem

Para o início do desenvolvimento deste trabalho foram selecionados inicialmente amostras de vídeos. Os vídeos utilizados estão disponíveis publicamente e fazem parte de uma coletânea amplamente utilizada em pesquisas relacionadas à transmissão e avaliação de conteúdo multimídia (ASU, 2014)(XIPH.ORG, 2014). Os critérios de escolha dos vídeos foram baseadas na estrutura de movimentação e formato das imagens. Três padrões foram aplicados nesta dissertação, indicados respectivamente nas Figuras 4.2, 4.3 e 4.4. As principais características dos vídeos usados são apresentados na Tabela 4.1.



(A)

Figura 4.2: Vídeo QCIF (*Quarter Common Intermediate Format*) utilizado: (A) Highway.



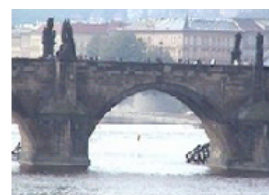
(A)



(B)



(C)



(D)

Figura 4.3: Vídeos CIF (*Common Intermediate Format*) utilizados: (A) Akiyo, (B) Coastguard, (C) Football e (D) Bridge Close.



(A)



(B)



(C)

Figura 4.4: Vídeos HD utilizados: (A) Big Buck Bunny, (B) Crown Run e (C) Blue Sky.

A preparação do vídeo para uso se inicia na codificação do vídeo. Os vídeos em formato YUV são codificados com H.264 utilizando a ferramenta FFMPEG (COUDU-



Tabela 4.1: Características principais dos vídeos SD e HD utilizados para simulação e testes.

Vídeo	Quadros	Tempo (seg)	Formato	Resolução	fps
<i>Highway</i>	2000	67	QCIF	176x144	30
<i>Akiyo</i>	300	11	CIF	352x288	25
<i>Coastguard</i>	300	11	CIF	352x288	25
<i>Football</i>	130	10	CIF	352x288	25
<i>Bridge Close</i>	2001	80	CIF	352x288	25
<i>Big Buck Bunny</i>	721	30	HD	1920x1080	24
<i>Crowd Run</i>	500	20	HD	1920x1080	24
<i>Blue Sky</i>	217	9	HD	1920x1080	24

RIER et al., 2015). Inicialmente o vídeo foi compactado a partir do padrão original do vídeo QCIF, CIF ou HD, a uma taxa de frequência do vídeo em quadros por segundo de 25 para CIF e 24 para HD, seleção do codec H.264 e tamanho do GoP de 12 quadros para todos os vídeos. O vídeo Highway é codificado a uma taxa diferente devido ao fato de que este mesmo vídeo é usado nos resultados do trabalho de (XU et al., 2011) e serão comparados posteriormente. O resultado desta compactação é uma versão do vídeo armazenado em arquivo com a extensão .m4v.

Na etapa seguinte o vídeo compactado (.m4v) é utilizado para criação do *hint track* pela ferramenta *MP4Box*. O *hint track* configura para qual padrão de protocolo de transmissão os pacotes serão usados. Foi utilizado para este experimento o protocolo RTP (*Real-Time Transport Protocol*) (SCHULZRINNE; CASNER; FREDERICK, 2003). O *MP4Box* resulta no arquivo de extensão .mp4.

Com a ferramenta *MP4Trace* é realizada a transmissão do vídeo. O próprio programa faz o envio do vídeo via protocolo RTP/UDP, para um endereço IP e porta de destino. Como resultado, é gerado um arquivo contendo a lista completa dos quadros gerados especificando o tipo, o tamanho em bytes, a quantidade de pacotes gerados e o tempo da geração do quadro, e também um segundo arquivo com a lista de todos os pacotes gerados com a informação do tipo de quadro ao qual pertence o pacote, o instante de tempo da geração e o tamanho em bytes do pacote.

## 4.2.2 Transmissão e Simulação

Foram coletadas amostras de atraso em redes de acesso do tipo ADSL, 3G e WIFI. Este cenário de possibilidades de conexão à Internet foram escolhidos por representarem opções tipicamente disponíveis ao usuário. A rede ADSL em teste é operada pela GVT (*Global Village Telecom*), a rede 3G é operada pela Claro. A rede WIFI é a rede do Departamento de Engenharia Elétrica da UFPR, que utiliza o padrão IEEE802.11g e está conectada à uma rede Ethernet não congestionada, que por sua vez conecta-se com a Internet. A conexão da UFPR com a Internet utiliza enlace com taxa de 20Gbps. O mesmo trace de atrasos foi empregado em (TORRES, 2014).

Em todos os casos, os dados foram coletados em horários comerciais para refletir o uso regular do sistema. O atraso de propagação foi obtido monitorando o envio de pacotes ICMP (*Internet Control Message Protocol*) ECHO REQUEST/REPLY utilizando o programa *ping*, com um computador conectado na rede de acesso. O servidor escolhido foi 198.45.48.71 (*NetFlix*).

Também foi gerado um trace de atrasos a partir da recomendação G.1050, *Network model for evaluating multimedia transmission performance over internet protocol* (ITU-T, 2007b). Nesta recomendação é descrito um modelo de geração de atrasos a partir de séries temporais e respectiva configuração para situações com diversos níveis de qualidade de serviço. O código fonte é apresentado no Apêndice A.

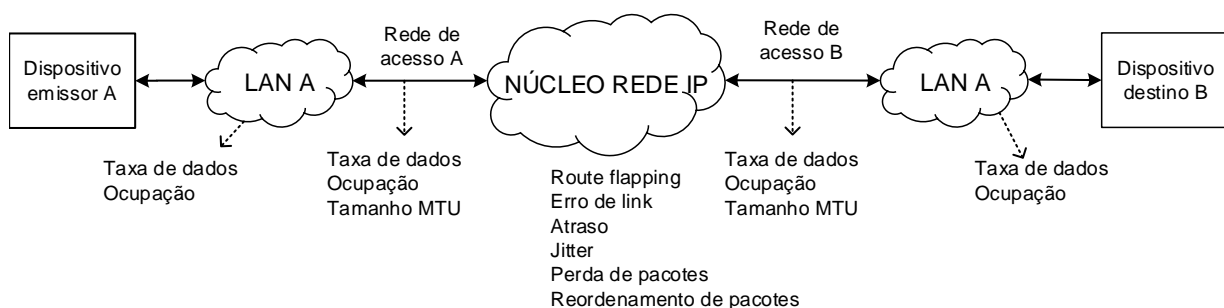


Figura 4.5: Modelo de rede utilizado pela recomendação G.1050.

A recomendação G.1050 apresenta um modelo para geração do atraso a partir

de parâmetros de configuração também disponíveis na recomendação. A Figura 4.5 apresenta o modelo de rede utilizado na recomendação G.1050. O modelo consiste em dois nós, A e B, conectados através de 5 redes: LAN A, REDE DE ACESSO A, NÚCLEO, REDE DE ACESSO B e LAN B. Em cada rede é indicado os parâmetros de configuração possíveis para alteração, estes parâmetros são indicados abaixo de cada rede e suas constantes se encontram disponíveis na recomendação. O modelo sugere o uso de diversos padrões de severidade de perdas e atrasos, denominados A, B, C, D, E, F, G e H, respectivamente do melhor para o pior desempenho. A recomendação G.1050 utiliza o modelo de Gilbert-Elliot (EBERT; WILLIG et al., 1999) para perda de pacotes e também para estabelecer as condições de atraso da rede. O atraso é modelado por uma média móvel exponencial.

#### 4.2.2.1 Transmissão

O arquivo com os pacotes de vídeo fornecidos pelo *MP4Trace* geram tempos de criação de cada pacote a partir do codificador. Contudo, os pacotes não são enviados no mesmo instante de tempo devido à ocupação do meio de transmissão. Assim, foi implementado um *script* em *bash* que gera os tempos corretos para cada pacote de vídeo considerando um canal de transmissão de 100 Mbps. Os tempos de geração dos pacotes de vídeo criados são acrescidos dos tempos de atraso das redes utilizadas 3G, WIFI, ADSL e pelo modelo da recomendação G.1050.

#### 4.2.3 Remontagem e Avaliação

O recebimento dos pacotes de vídeo pode ocorrer nos dois caminhos, primário e secundário. No caminho primário o vídeo é enviado de modo completo e pelo canal secundário são recebidos os pacotes redundantes decorrente do percentual de redundância da etapa de transmissão anterior.

No processo de remontagem do arquivo recebido, os pacotes enviados pelo canal

secundário são comparados ao respectivo (ID) pacote recebido no caminho primário. Entre os pacotes iguais recebidos é mantido o de menor tempo e o de maior tempo é descartado.

Com a remontagem do arquivo dos pacotes recebidos é feita a reconstrução do vídeo recebido. A reconstrução é feita com o auxílio do *Evaluate Traces of MP4-file transmission (Etmp4)* que gera um vídeo de formato H.264. O *Etmp4* utiliza dos arquivos de envio, recebimento e um *playout buffer* de tamanho configurável. O *playout buffer* foi configurado com valor fixo de 200 ms, pacotes com atraso maior a este valor são considerados como perdidos e não são usados na remontagem. Posteriormente é usado o FFMPEG, para converter o vídeo H.264 recebido para o formato YUV, que será usado para avaliação da qualidade através das métricas objetivas.

Os vídeos em formato YUV recebidos são aplicados na comparação de qualidade utilizando o PSNR e SSIM, comparando-se o vídeo original e o recebido e assim permitindo avaliar o nível de degradação do vídeo transmitido. Os resultados de PSNR e SSIM são apresentados com um erro estimado a partir de intervalos de confiança.

O intervalo de confiança é obtido através do modelo de confiança *t-student*. O semi intervalo  $h$  onde se encontra a verdadeira média, com um nível desejado de confiança, pode ser encontrado calculando-se  $h = t_{n-1, 1-\alpha/2} \frac{\sigma}{\sqrt{n}}$ , onde  $n$  é o número de rodadas,  $\alpha$  é o nível de significância,  $\sigma$  é o desvio padrão,  $1 - \alpha/2$  é o nível de confiança e  $t$  indica o valor crítico para distribuição *t-student*.

## CAPÍTULO 5

### ANÁLISE DOS RESULTADOS E DISCUSSÃO

A realização e apresentação dos resultados foram feitos a partir de um conjunto de simulações repetidas quarenta vezes para cada vídeo com intervalos de atrasos diferentes com intervalo de confiança de 90%. São utilizadas diferentes configurações de caminho primário e secundário. Os resultados são demonstrados em três etapas: na primeira etapa são apresentados os resultados obtidos com a transmissão redundante usando o método de redundância relativa aplicada às funções de mapeamento linear, exponencial e quadrática; na segunda etapa são apresentados os resultados na transmissão com método preditivo; e por último são indicados os resultados com variação de variáveis de desempenho do método.

O ganho de qualidade é observado a partir da diferença obtida do resultado do PSNR, na transmissão usando o método RSTM com caminhos secundários pelo resultado obtido na transmissão *unicast* sem redundância.

#### 5.1 Transmissão *bicasting* de vídeo

Primeiramente, na Tabela 5.1 são mostrados os valores máximos de PSNR e SSIM possíveis para as simulações realizadas. Este valor representa a transmissão do vídeo na rede e remontagem total do vídeo recebido sem perda de nenhum pacote no receptor. A degradação de qualidade percebida é resultado da compactação do H.264.

Considerando um sistema com 100% de redundância conforme proposto por (TARABUTA; POTORAC; BALAN, 2013), em que é enviado o mesmo vídeo pelo caminho secundário, foram obtidos os valores médios do PSNR e SSIM indicados na Tabela 5.2. Foram realizadas transmissões simuladas utilizando as amostras de atraso da

Tabela 5.1: Níveis máximos de PSNR e SSIM possíveis para vídeos testados.

Vídeo	PSNR	SSIM
<i>Highway</i>	38,47	0,93
<i>Akiyo</i>	43,99	0,99
<i>Coastguard</i>	37,27	0,96
<i>Football</i>	37,54	0,96
<i>Bridge Close</i>	37,21	0,93
<i>Big Buck Bunny</i>	45,98	0,99
<i>Crown Run</i>	35,01	0,94
<i>Blue Sky</i>	40,07	0,97

rede WIFI, seguida pelas combinações WIFI+3G e WIFI+ADSL, representando respectivamente o caminho primário e secundário. A abordagem do uso da rede WIFI como caminho primário de transmissão decorre do fato no uso crescente de dispositivos com interface sem fio.

O uso da transmissão com 100% de redundância sempre resultou em um ganho de qualidade, como indicado na Tabela 5.2 na coluna do ganho. Em ambos os casos, WIFI+3G e WIFI+ADSL, é possível verificar que os ganhos de PSNR foram significativos permitindo ganhos reais a partir de 3,2 dB em todos os vídeos analisados. É possível identificar que os ganhos usando a rede ADSL como caminho secundário resultou em ganhos maiores se comparado ao da rede 3G como secundário, com uma média de melhoria de 3,91 dB para rede ADSL e 3,75 dB para rede 3G como secundário.

O ganho de qualidade observado com a transmissão com 100% de redundância indicam que o uso de sistemas multiabrigados pode ser de grande auxílio na melhoria de qualidade percebida pelo usuário. Contudo, os resultados indicados na Tabela 5.2 exigem a duplicação de tráfego pelo caminho secundário, o que pode se tornar inviável e desnecessário na maioria dos casos.

Na próxima seção são apresentados os resultados da transmissão com redundância variável do método relativo. O objetivo é diminuir o percentual de redundância e consequentemente o uso do caminho secundário.

Tabela 5.2: Qualidade obtida a partir da transmissão 100% de redundância de vídeo no caminho secundário em 40 simulações e intervalo de confiança *t-student* 90%.

Vídeo		PSNR(dB)	SSIM	Ganho(dB)
Highway	WIFI sem redundância	$34,71 \pm 0,70$	$0,884 \pm 0,009$	
	WIFI + 3G	$38,36 \pm 0,09$	$0,932 \pm 0,001$	3,65
	WIFI + ADSL	$38,40 \pm 0,04$	$0,932 \pm 0,001$	3,69
Akiyo	WIFI sem redundância	$39,38 \pm 1,24$	$0,931 \pm 0,015$	
	WIFI + 3G	$43,45 \pm 0,29$	$0,980 \pm 0,003$	4,07
	WIFI + ADSL	$43,73 \pm 0,16$	$0,983 \pm 0,002$	4,35
Coastguard	WIFI sem redundância	$33,65 \pm 0,97$	$0,881 \pm 0,023$	
	WIFI + 3G	$36,84 \pm 0,21$	$0,955 \pm 0,005$	3,20
	WIFI + ADSL	$37,03 \pm 0,12$	$0,959 \pm 0,003$	3,39
Football	WIFI sem redundância	$33,73 \pm 1,02$	$0,870 \pm 0,024$	
	WIFI + 3G	$37,19 \pm 0,19$	$0,951 \pm 0,005$	3,46
	WIFI + ADSL	$37,37 \pm 0,15$	$0,955 \pm 0,003$	3,64
Briedge Close	WIFI sem redundância	$33,92 \pm 0,72$	$0,878 \pm 0,011$	
	WIFI + 3G	$37,14 \pm 0,05$	$0,925 \pm 0,001$	3,31
	WIFI + ADSL	$37,16 \pm 0,03$	$0,925 \pm 0,001$	3,33
Big Buck Bunny	WIFI sem redundância	$41,77 \pm 1,15$	$0,950 \pm 0,010$	
	WIFI + 3G	$45,68 \pm 0,19$	$0,987 \pm 0,001$	3,91
	WIFI + ADSL	$45,84 \pm 0,09$	$0,988 \pm 0,001$	4,07
Crown Run	WIFI sem redundância	$31,19 \pm 1,04$	$0,834 \pm 0,005$	
	WIFI + 3G	$34,70 \pm 0,21$	$0,931 \pm 0,005$	3,51
	WIFI + ADSL	$34,80 \pm 0,13$	$0,933 \pm 0,003$	3,61
Blue Sky	WIFI sem redundância	$34,63 \pm 1,42$	$0,882 \pm 0,022$	
	WIFI + 3G	$39,49 \pm 0,34$	$0,963 \pm 0,006$	4,86
	WIFI + ADSL	$39,83 \pm 0,19$	$0,969 \pm 0,003$	5,20

## 5.2 Transmissão em Redundância relativa

Inicialmente foi analisado o efeito da variação da constante  $\alpha$  no desempenho do método. Os resultados são apresentados na Tabela 5.3. Testes realizados com o vídeo Akyio utilizando WIFI como caminho primário e ADSL como caminho secundário mostram que com  $\alpha = 0,9$  a quantidade média de informação redundante foi de 21,10% do total e 23,17%, 23,96%, 23,04% e 21,71%, respectivamente, para  $\alpha$  igual a 1/2, 1/4, 1/6 e 1/8. Esta mesma abordagem é observada no ganho de qualidade, onde também com  $\alpha = 0,9$  obteve-se uma melhora de em média 3,65 dB, seguidos por 3,51 dB, 3,26 dB, 2,97 dB e 2,76 dB, respectivamente para os valores de  $\alpha$  de 1/2, 1/4, 1/6 e 1/8. O desempenho da transmissão de outros vídeos é semelhante, bem como se for usado como caminho secundário a rede 3G. A partir desses resultados,

Tabela 5.3: Resultados obtidos para  $\alpha = 0,9$ ,  $\alpha = 1/2$ ,  $\alpha = 1/4$ ,  $\alpha = 1/6$  e  $\alpha = 1/8$  combinados aos métodos relativos Linear, Exponencial e Quadrático para vídeo Akiyo.

$\alpha$	Transmissão	PSNR	SSIM	Ganho(dB)	Redundância
$\alpha = 0,9$	WIFI sem redundância	$39,38 \pm 1,25$	$0,931 \pm 0,015$		
	WIFI + 3G	Linear	$42,83 \pm 0,39$	3,45	26,79%
		Exponencial	$42,65 \pm 0,42$	3,27	16,13%
		Quadrático	$42,72 \pm 0,42$	3,34	20,39%
	WIFI + ADSL	Linear	$43,08 \pm 0,24$	3,70	26,79%
		Exponencial	$42,94 \pm 0,26$	3,56	16,13%
		Quadrático	$43,07 \pm 0,24$	3,69	20,39%
$\alpha = 1/2$	WIFI + 3G	Linear	$42,95 \pm 0,32$	3,57	30,14%
		Exponencial	$42,59 \pm 0,38$	3,21	16,78%
		Quadrático	$42,80 \pm 0,33$	3,42	22,60%
	WIFI + ADSL	Linear	$43,09 \pm 0,23$	3,71	30,14%
		Exponencial	$42,64 \pm 0,32$	3,26	16,78%
		Quadrático	$42,94 \pm 0,25$	3,56	22,60%
$\alpha = 1/4$	WIFI + 3G	Linear	$42,74 \pm 0,35$	3,36	31,27%
		Exponencial	$42,09 \pm 0,49$	2,71	17,26%
		Quadrático	$42,42 \pm 0,42$	3,04	23,35%
	WIFI + ADSL	Linear	$42,94 \pm 0,28$	3,56	31,27%
		Exponencial	$42,26 \pm 0,46$	2,88	17,26%
		Quadrático	$42,73 \pm 0,34$	3,35	23,35%
$\alpha = 1/6$	WIFI + 3G	Linear	$42,51 \pm 0,41$	3,13	30,57%
		Exponencial	$41,75 \pm 0,58$	2,37	16,10%
		Quadrático	$42,15 \pm 0,48$	2,77	22,46%
	WIFI + ADSL	Linear	$42,75 \pm 0,34$	3,37	30,57%
		Exponencial	$41,93 \pm 0,55$	2,55	16,10%
		Quadrático	$42,37 \pm 0,46$	2,99	22,46%
$\alpha = 1/8$	WIFI + 3G	Linear	$42,26 \pm 0,48$	2,88	29,49%
		Exponencial	$41,41 \pm 0,71$	2,03	14,47%
		Quadrático	$41,83 \pm 0,60$	2,45	21,16%
	WIFI + ADSL	Linear	$42,53 \pm 0,42$	3,15	29,49%
		Exponencial	$41,68 \pm 0,69$	2,30	14,47%
		Quadrático	$42,20 \pm 0,54$	2,82	21,16%

foi escolhido o valor de  $\alpha$  de 0,9 para uso nos testes subsequentes.

As Tabelas 5.4 e 5.5 apresentam os valores médios obtidos para as métricas de PSNR e SSIM, respectivamente para vídeos SD e HD, bem como o erro calculado com o intervalo de confiança de 90%. Pode-se observar que em ambas as tabelas houve melhora da qualidade medida pelo PSNR e SSIM de todos os vídeos simulados. Quando comparado à transmissão sem redundância usando somente o caminho WIFI e a transmissão redundante usando dois canais distintos são obtidos ganhos de PSNR a partir de 2.5 dB para o vídeo *Coastguard* e ganho máximo de 3.5 dB no caso do vídeo em HD Blue Sky. Como já esperado, a comparação entre o uso do 3G e o



ADSL como canais secundário de transmissão redundante apresentaram diferenças entre seus ganhos. Em todos os casos os ganhos usando o ADSL foram novamente maiores quando usando a mesma função de mapeamento.

A transmissão de todos os quadros I de forma redundante também é estudada e demonstrou um ganho de PSNR positivo. Este estudo de desempenho é uma das contribuições desta dissertação. Contudo, a transmissão redundante de todos os quadros I não foi fator decisivo para garantir melhores ganhos. É possível observar que alguns ganhos como no caso dos vídeos Football e Big Buck Bunny são próximos a zero. Isto ocorre em decorrência da compactação realizada pelo MPEG, sendo que nestes dois vídeos especificamente o alto nível de movimentação faz com que a decodificação do vídeo seja muito dependente dos quadros P.

As diferentes funções de mapeamento de cálculo de redundância permitiram a redução da quantidade de informação redundante enviada pelo caminho secundário. Diferente do método de transmissão 100% de redundância é possível observar que os percentuais reduziram em média de 26,64% para o mapeamento utilizando a função linear e para 20,21% e 16,04%, respectivamente para função quadrática e exponencial. Em todos os casos, observa-se que a função de mapeamento linear apresentou maiores ganhos de qualidade, porém com maior envio de redundância que as funções de mapeamento concorrentes. Em contrapartida, a função exponencial apresentou os menores ganhos que as demais funções porém com menor uso de redundância. Ao realizar a relação entre os ganhos obtidos com as funções de mapeamento em razão do ganho obtido na transmissão 100% pode-se verificar que: as médias de ganho das funções linear, quadrática e exponencial representam aproximadamente 85%, 82% e 79% da média total da transmissão *bicasting*.

Assim, pode-se dizer que em relação a ganhos de qualidade o melhor desempenho foi obtido pela função linear seguida da função quadrática e por último função exponencial. Em relação a percentuais de redundância os menores percentuais de redundância são obtidos através da função exponencial seguida pela função quadrática

Tabela 5.4: Médias obtidas de PSNR e SSIM para transmissão redundante relativa para vídeos SD com  $\alpha = 0,9$ .

Vídeo	Transmissão/Método	PSNR	SSIM	Ganho(dB)	Redundância
Highway	WIFI sem redundância	$34,71 \pm 0,70$	$0,844 \pm 0,009$		
	Red. Total	$38,36 \pm 0,09$	$0,932 \pm 0,001$	3,65	100,0%
	Quadro I	$37,42 \pm 0,21$	$0,921 \pm 0,003$	2,71	51,00%
	WIFI + 3G				
	Linear	$37,97 \pm 0,12$	$0,927 \pm 0,001$	3,26	26,48%
	Exponencial	$37,74 \pm 0,14$	$0,924 \pm 0,002$	3,03	16,19%
	Quadrático	$37,86 \pm 0,13$	$0,926 \pm 0,002$	3,15	20,15%
	Red. Total	$38,40 \pm 0,04$	$0,932 \pm 0,001$	3,69	100,0%
	Quadro I	$37,45 \pm 0,20$	$0,921 \pm 0,003$	2,74	51,00%
	WIFI + ADSL				
	Linear	$37,99 \pm 0,10$	$0,927 \pm 0,001$	3,28	26,48%
	Exponencial	$37,79 \pm 0,13$	$0,925 \pm 0,002$	3,08	16,19%
	Quadrático	$37,90 \pm 0,11$	$0,926 \pm 0,001$	3,19	20,15%
Akiyo	WIFI sem redundância	$39,38 \pm 1,25$	$0,931 \pm 0,015$		
	Red. Total	$43,45 \pm 0,29$	$0,980 \pm 0,003$	4,07	100,0%
	Quadro I	$42,23 \pm 0,51$	$0,974 \pm 0,005$	2,85	80,15%
	WIFI + 3G				
	Linear	$42,83 \pm 0,39$	$0,975 \pm 0,005$	3,45	26,79%
	Exponencial	$42,65 \pm 0,42$	$0,972 \pm 0,005$	3,27	16,13%
	Quadrático	$42,72 \pm 0,42$	$0,973 \pm 0,005$	3,34	20,39%
	Red. Total	$43,73 \pm 0,16$	$0,983 \pm 0,002$	4,35	100,0%
	Quadro I	$42,44 \pm 0,41$	$0,977 \pm 0,003$	3,06	80,15%
	WIFI + ADSL				
	Linear	$43,08 \pm 0,24$	$0,978 \pm 0,002$	3,70	26,79%
	Exponencial	$42,94 \pm 0,26$	$0,977 \pm 0,003$	3,56	16,13%
	Quadrático	$43,07 \pm 0,24$	$0,979 \pm 0,003$	3,69	20,39%
Coastguard	WIFI sem redundância	$33,64 \pm 0,97$	$0,881 \pm 0,023$		
	Red. Total	$36,84 \pm 0,21$	$0,955 \pm 0,005$	3,20	100,0%
	Quadro I	$34,87 \pm 0,64$	$0,907 \pm 0,015$	1,23	37,76%
	WIFI + 3G				
	Linear	$36,32 \pm 0,30$	$0,944 \pm 0,007$	2,68	26,79%
	Exponencial	$36,14 \pm 0,34$	$0,940 \pm 0,008$	2,50	16,13%
	Quadrático	$36,23 \pm 0,32$	$0,942 \pm 0,007$	2,59	20,39%
	Red. Total	$37,03 \pm 0,12$	$0,959 \pm 0,003$	3,39	100,0%
	Quadro I	$34,95 \pm 0,61$	$0,909 \pm 0,025$	1,31	37,76%
	WIFI + ADSL				
	Linear	$36,43 \pm 0,21$	$0,946 \pm 0,005$	2,79	26,79%
	Exponencial	$36,28 \pm 0,24$	$0,943 \pm 0,005$	2,64	16,13%
	Quadrático	$36,39 \pm 0,22$	$0,945 \pm 0,005$	2,75	20,39%
Football	WIFI sem redundância	$33,73 \pm 1,02$	$0,870 \pm 0,024$		
	Red. Total	$37,19 \pm 0,19$	$0,951 \pm 0,005$	3,46	100,0%
	Quadro I	$34,08 \pm 0,94$	$0,879 \pm 0,021$	0,35	24,40%
	WIFI + 3G				
	Linear	$36,59 \pm 0,28$	$0,938 \pm 0,006$	2,86	30,24%
	Exponencial	$36,41 \pm 0,33$	$0,933 \pm 0,008$	2,68	18,57%
	Quadrático	$36,52 \pm 0,30$	$0,936 \pm 0,007$	2,79	23,49%
	Red. Total	$37,37 \pm 0,15$	$0,955 \pm 0,003$	3,64	100,0%
	Quadro I	$34,10 \pm 0,93$	$0,881 \pm 0,021$	0,37	24,40%
	WIFI + ADSL				
	Linear	$36,77 \pm 0,23$	$0,943 \pm 0,005$	3,04	30,24%
	Exponencial	$36,53 \pm 0,28$	$0,936 \pm 0,006$	2,80	18,57%
	Quadrático	$36,70 \pm 0,24$	$0,941 \pm 0,005$	2,97	23,49%
Bridge Close	WIFI sem redundância	$33,83 \pm 0,74$	$0,876 \pm 0,011$		
	Red. Total	$37,14 \pm 0,05$	$0,925 \pm 0,000$	3,31	100,0%
	Quadro I	$36,62 \pm 0,13$	$0,919 \pm 0,001$	2,79	53,67%
	WIFI + 3G				
	Linear	$36,80 \pm 0,08$	$0,921 \pm 0,001$	2,97	24,91%
	Exponencial	$36,57 \pm 0,12$	$0,917 \pm 0,002$	2,74	14,60%
	Quadrático	$36,71 \pm 0,10$	$0,919 \pm 0,001$	2,88	18,49%
	Red. Total	$37,16 \pm 0,03$	$0,925 \pm 0,000$	3,33	100,0%
	Quadro I	$36,64 \pm 0,12$	$0,920 \pm 0,001$	2,81	53,67%
	WIFI + ADSL				
	Linear	$36,81 \pm 0,08$	$0,921 \pm 0,001$	2,98	24,91%
	Exponencial	$36,57 \pm 0,13$	$0,917 \pm 0,002$	2,74	14,60%
	Quadrático	$36,72 \pm 0,10$	$0,919 \pm 0,001$	2,89	18,49%

Tabela 5.5: Médias obtidas de PSNR e SSIM para transmissão redundante relativa para vídeos HD com  $\alpha = 0,9$ .

Vídeo	Transmissão/Método	PSNR	SSIM	Ganho(dB)	Redundância
Big Buck Bunny	WIFI sem redundância	41,77 $\pm$ 1,15	0,950 $\pm$ 0,010		
	Red. Total	45,68 $\pm$ 0,19	0,987 $\pm$ 0,001	3,91	100,0%
	Quadro I	41,94 $\pm$ 1,13	0,951 $\pm$ 0,011	0,17	69,88%
	WIFI + 3G				
	Linear	44,93 $\pm$ 0,29	0,981 $\pm$ 0,003	3,16	22,23%
	Exponencial	44,73 $\pm$ 0,33	0,979 $\pm$ 0,003	2,96	12,91%
	Quadrático	44,82 $\pm$ 0,30	0,979 $\pm$ 0,003	3,05	16,26%
	Red. Total	45,84 $\pm$ 0,09	0,988 $\pm$ 0,001	4,07	100,0%
	Quadro I	41,95 $\pm$ 1,13	0,952 $\pm$ 0,012	0,18	69,88%
	WIFI + ADSL				
	Linear	45,10 $\pm$ 0,19	0,982 $\pm$ 0,002	3,33	22,23%
	Exponencial	44,89 $\pm$ 0,24	0,980 $\pm$ 0,002	3,12	12,91%
	Quadrático	45,00 $\pm$ 0,21	0,981 $\pm$ 0,002	3,23	16,26%
Crown Run	WIFI sem redundância	31,19 $\pm$ 1,04	0,834 $\pm$ 0,028		
	Red. Total	34,70 $\pm$ 0,21	0,931 $\pm$ 0,005	3,51	100,0%
	Quadro I	32,36 $\pm$ 0,75	0,854 $\pm$ 0,024	1,17	35,94%
	WIFI + 3G				
	Linear	34,18 $\pm$ 0,28	0,917 $\pm$ 0,007	2,99	25,59%
	Exponencial	33,95 $\pm$ 0,32	0,910 $\pm$ 0,008	2,76	15,37%
	Quadrático	34,10 $\pm$ 0,30	0,914 $\pm$ 0,008	2,91	19,28%
	Red. Total	34,80 $\pm$ 0,13	0,933 $\pm$ 0,003	3,61	100,0%
	Quadro I	32,40 $\pm$ 0,72	0,855 $\pm$ 0,023	1,21	35,94%
	WIFI + ADSL				
	Linear	34,26 $\pm$ 0,19	0,918 $\pm$ 0,005	3,07	25,59%
	Exponencial	34,02 $\pm$ 0,23	0,912 $\pm$ 0,006	2,83	15,37%
	Quadrático	34,17 $\pm$ 0,20	0,916 $\pm$ 0,006	2,98	19,28%
Blue Sky	WIFI sem redundância	34,63 $\pm$ 1,42	0,882 $\pm$ 0,024		
	Red. Total	39,49 $\pm$ 0,34	0,963 $\pm$ 0,006	4,86	100,0%
	Quadro I	36,12 $\pm$ 1,07	0,898 $\pm$ 0,021	1,49	53,79%
	WIFI + 3G				
	Linear	38,55 $\pm$ 0,50	0,949 $\pm$ 0,008	3,92	30,10%
	Exponencial	38,22 $\pm$ 0,56	0,942 $\pm$ 0,010	3,59	18,40%
	Quadrático	38,40 $\pm$ 0,52	0,945 $\pm$ 0,008	3,77	23,26%
	Red. Total	39,83 $\pm$ 0,19	0,969 $\pm$ 0,003	5,20	100,0%
	Quadro I	36,22 $\pm$ 1,02	0,899 $\pm$ 0,020	1,59	53,79%
	WIFI + ADSL				
	Linear	38,80 $\pm$ 0,32	0,953 $\pm$ 0,005	4,17	30,10%
	Exponencial	38,48 $\pm$ 0,40	0,947 $\pm$ 0,006	3,85	18,40%
	Quadrático	38,65 $\pm$ 0,35	0,950 $\pm$ 0,006	4,02	23,26%

e por último função linear.

O uso da função de mapeamento linear resultou em maior transmissão de redundância se comparado as funções quadrática e exponencial. Neste panorama, o modelo exponencial se tornou a melhor alternativa ao enviar menos redundância garantindo níveis de qualidade bastante próximos do apresentado pelo modelo linear.

### 5.2.1 Transmissão com tempos de atraso gerador G.1050

Os resultados apresentados nesta subseção são decorrentes das simulações realizadas a partir dos atrasos gerados de acordo com a recomendação G.1050 (ITU-

T, 2007b). São usadas as configurações de rede descritas na Tabela 5.6 com redundância relativa usando a função de mapeamento exponencial. O uso desta recomendação para geração de atrasos a partir de diferentes padrões de degradação de rede permite fazer uma avaliação mais geral do desempenho do método RSTM.

Tabela 5.6: Configurações de rede usados para gerador de atrasos baseado na recomendação G.1050.

Rede	Padrão de degradação de qualidade
<i>LAN A</i>	A
<i>Rede de acesso A</i>	A, B, C, D, E, F, G e H
<i>Núcleo Rede IP</i>	C
<i>Rede de acesso B</i>	A, B, C, D, E, F, G e H
<i>LAN B</i>	A

Para cada conjunto de testes foram gerados dois *traces* de atraso. O objetivo é testar configurações de rede que apresentam perfil de degradação similar entre o caminho primário e secundário, com uma deterioração progressiva das condições de qualidade de serviço. A notação usada para representação da configuração de rede utilizada, leva em consideração somente a alteração das redes de *Rede de acesso A* e *B*. Os demais parâmetros são iguais respectivamente para *LAN A*, *Núcleo Rede IP* e *LAN B*, como indicado na Tabela 5.6.

A Tabela 5.7 apresenta os resultados da simulação do vídeo Highway em formato CIF. Mantendo-se a configuração da rede de acesso A fixa no padrão A e variando a configuração da rede de acesso B de A a H pode-se observar que a variação do padrão de qualidade de serviço leva a uma degradação de qualidade, onde redes que utilizam a configuração A possuem maiores valores médios de PSNR, que diminui gradativamente até o padrão de configuração H. Essa degradação ocorre devido aos tempos de atrasos da rede H serem maiores. O aumento dos tempos de atraso da rede com configuração A para H também reflete no percentual de redundância, começando em A com 0,06% até H com um valor máximo possível de 20,35%. O ganho na transmissão redundante é perceptível a partir do uso do padrão F com um ganho de 0,05 dB. A utilização de redes com padrões piores como G e H, geram maiores médias de ganho de PSNR decorrente da compensação do uso do método RSTM. A utilização

Tabela 5.7: Resultados obtidos para modelo gerador de atrasos, com *rede de acesso A e B* variável, recomendação G.1050 com  $\alpha = 0,9$  e 95% de confiança para vídeo Highway.

Transmissão	PSNR	SSIM	Ganho(dB)	Redundância
Sem redundância	$38,98 \pm 0,04$	$0,927 \pm 0,000$		
AA+C+AA	$38,98 \pm 0,04$	$0,927 \pm 0,000$	0,00	0,06%
Sem redundância	$38,92 \pm 0,05$	$0,927 \pm 0,000$		
AA+C+BA	$38,92 \pm 0,05$	$0,927 \pm 0,000$	0,00	0,06%
Sem redundância	$38,57 \pm 0,07$	$0,924 \pm 0,001$		
AA+C+CA	$38,57 \pm 0,07$	$0,924 \pm 0,001$	0,00	0,12%
Sem redundância	$38,27 \pm 0,09$	$0,921 \pm 0,001$		
AA+C+DA	$38,27 \pm 0,09$	$0,921 \pm 0,001$	0,00	0,18%
Sem redundância	$37,74 \pm 0,12$	$0,916 \pm 0,001$		
AA+C+EA	$37,74 \pm 0,12$	$0,916 \pm 0,001$	0,00	0,29%
Sem redundância	$36,90 \pm 0,13$	$0,909 \pm 0,001$		
AA+C+FA	$36,95 \pm 0,13$	$0,909 \pm 0,001$	0,05	0,72%
Sem redundância	$34,64 \pm 0,21$	$0,888 \pm 0,002$		
AA+C+GA	$35,17 \pm 0,19$	$0,893 \pm 0,002$	0,53	4,58%
Sem redundância	$31,81 \pm 0,26$	$0,861 \pm 0,002$		
AA+C+HA	$34,36 \pm 0,16$	$0,885 \pm 0,002$	2,55	20,35%
Sem redundância	$31,78 \pm 0,24$	$0,863 \pm 0,002$		
AB+C+HA	$34,50 \pm 0,17$	$0,886 \pm 0,002$	2,72	24,76%
Sem redundância	$31,71 \pm 0,23$	$0,859 \pm 0,002$		
AC+C+HA	$34,76 \pm 0,19$	$0,889 \pm 0,002$	3,05	27,37%
Sem redundância	$31,36 \pm 0,22$	$0,855 \pm 0,002$		
AD+C+HA	$34,51 \pm 0,18$	$0,886 \pm 0,002$	3,15	28,28%
Sem redundância	$30,73 \pm 0,26$	$0,849 \pm 0,003$		
AE+C+HA	$34,18 \pm 0,18$	$0,883 \pm 0,002$	3,45	30,01%
Sem redundância	$29,88 \pm 0,23$	$0,841 \pm 0,002$		
AF+C+HA	$34,20 \pm 0,17$	$0,883 \pm 0,002$	4,32	37,62%
Sem redundância	$28,13 \pm 0,24$	$0,823 \pm 0,002$		
AG+C+HA	$33,81 \pm 0,20$	$0,880 \pm 0,002$	5,68	54,01%
Sem redundância	$25,99 \pm 0,25$	$0,801 \pm 0,003$		
AH+C+HA	$32,57 \pm 0,18$	$0,868 \pm 0,002$	6,56	70,86%

do padrão de severidade G amentou a média de ganho para 0,53 dB a uma média de redundância de 4,58% e para o padrão H com 2,55 dB e 20,35% de redundância. Ao manter a rede de acesso B fixa no padrão H e alterar o padrão da rede de acesso A verifica-se que ocorre uma maior degradação da qualidade do vídeo de 31,81 dB no padrão A para um valor de PSNR com 25,99 dB no padrão H. Aplicando-se o método proposto observa-se uma preservação da qualidade, acompanhada de um incremento na quantidade de informação redundante transmitido pelo caminho secundário.

## 5.2.2 Comparação com trabalhos correlatos

Nesta seção é comparado o trabalho em (XU et al., 2011) onde são apresentados resultados utilizando o SCTP padrão com diferentes políticas de retransmissão e troca de caminho, com os resultados aplicando o CMT sem e com *Partial Reliability* também com diferentes políticas de retransmissão.

### 5.2.2.1 SCTP padrão

A configuração do SCTP padrão usada por XU et al. é configurada usando o PMR=0 para troca de caminho agressiva (AF) e PMR=1 para troca de caminho menos agressiva (LAF). Os testes realizados pelos autores utilizam configurações de caminhos simétricos com percentuais fixos de perdas entre 1%, 3% e 5% para o teste de três políticas de retransmissão: AllRtxAlt, onde todas as retransmissões são realizadas por um caminho alternativo; AllRtxSame, com todas as retransmissões enviadas pelo mesmo caminho; e FrSameRtoAlt, com retransmissão rápida no mesmo caminho. É utilizado o vídeo Highway com resolução QCIF.

O método RSTM foi aplicado também ao vídeo Highway de formato QCIF para comparação entre os métodos. Os testes foram realizados utilizando os parâmetros indicados na Tabela 5.8.

Tabela 5.8: Configurações dos parâmetros para comparação dos resultados com SCTP padrão.

Parâmetro	Configuração
Método transmissão	Relativo
Função de mapeamento	Exponencial
$\alpha$	0,9
Gerador atrasos	Recomendação G.1050
Perdas de pacotes	1%, 3% e 5%

Os valores médios de PSNR, de 40 simulações, usando o método RSTM são apresentados na Tabela 5.9. É possível observar que o aumento do percentual de perda de pacotes acarretou na queda da qualidade medida pelo PSNR. Ao comparar os trabalhos, observou-se que utilizar a troca de caminho e retransmissão como no SCTP padrão apresentou um menor PSNR que na transmissão redundante. A diferença chegou a 2,39 dB, 2,9 dB e 5,17 dB respectivamente para 1%, 3% e 5%.

Tabela 5.9: Comparação resultados XU et al. SCTP padrão com método RSTM para o vídeo Highway QCIF.

Método		Transmissão	Perda 1%	Perda 3%	Perda 5%
(XU et al., 2011)			PSNR (dB)	PSNR (dB)	PSNR (dB)
	Aggresive Failover (AF)	AllRtxSame	35,60	34,10	31,70
		AllRtxAlt	35,65	34,41	32,01
		FrSameRtoAlt	35,60	34,10	29,90
	Less Aggresive Failover (LAF)	AllRtxSame	35,25	33,20	31,70
		AllRtxAlt	34,85	32,58	29,25
		FrSameRtoAlt	35,10	34,37	29,95
	Método RSTM	$\alpha = 0,9$	38,04	37,31	37,18

### 5.2.2.2 CMT padrão e PR-CMT

A configuração da utilização do CMT com e sem o *Partial Reliability* é apresentada a partir da utilização de diferentes políticas de retransmissão. Estes resultados são obtidos a partir da simulação em uma topologia de rede assimétrica com percentuais de perda de 4% no caminho primário e 8% no caminho secundário. O CMT padrão é apresentado por NPR e o PR-CMT com PR.

Novamente, o RSTM é utilizado com o vídeo Highway em formato QCIF utilizando

as características indicadas na Tabela 5.10.

Tabela 5.10: Configurações dos parâmetros para comparação dos resultados com CMT e PR-CMT.

Parâmetro	Configuração
Método transmissão	Relativo
Função de mapeamento	Exponencial
$\alpha$	0,9
Gerador atrasos	Recomendação G.1050
Perdas de pacotes caminho primário	4%
Perdas de pacotes caminho secundário	8%

Após testes, com 40 simulações, o resultado obtido com o método RSTM é apresentado na Tabela 5.11, assim como os resultados indicados com o uso do CMT padrão e o PR-CMT. Com um intervalo de confiança de 95%, o erro obtido de PSNR é de  $\pm 0,09$  dB com uma média de 37,61 dB. Este resultado é superior em relação a qualidade obtida no uso dos métodos CMT e PR-CMT.

Tabela 5.11: Comparação resultados XU et al. CMT e PR-CMT com método RSTM para o vídeo Highway QCIF.

Política de Retransmissão		Perda 4% e 8%
		PSNR (dB)
(XU et al., 2011)	ASAP	NPR
		PR0
		PR1
		PR2
	SAMA	NPR
		PR0
		PR1
		PR2
	SSTHRESH	NPR
		PR0
		PR1
		PR2
	LOSSRATE	NPR
		PR0
		PR1
		PR2
	CWND	NPR
		PR0
		PR1
		PR2
	Método RSTM	$\alpha = 0,9$

A próxima seção é destinada a aplicação dos sistemas de redundância relativa



associado ao modelo de redundância preditiva, que leva em consideração a tendência de queda ou subida do atraso suavizado na rede primária.

### 5.3 Transmissão redundância preditiva

O uso da estrutura de autocorrelação dos atrasos para antecipar possíveis problemas com a qualidade percebida tem o objetivo de reduzir o envio de informação redundante na rede secundária sem perder os resultados de qualidade já observados na transmissão de redundância relativa. Os resultados apresentados nesta seção são realizados todos sobre o modelo de mapeamento exponencial.

Inicialmente foi analisado o efeito da variação da constante  $\alpha$  no desempenho do método preditivo. Foram realizadas simulações com os valores de  $\alpha$  de  $1/2$ ,  $1/4$ ,  $1/6$  e  $1/8$  para análise dos resultados obtidos.

O modelo baseado no método de redundância preditiva utiliza de uma máquina de estados para definição de uma constante de multiplicação  $\rho$ , sendo necessário o uso da constante  $\psi$ . Foram testados valores de  $\psi$  de 1, 3, 5 e 7 e aplicados a dois vídeos diferentes, sendo um SD e outro em HD, respectivamente Coastguard e Big Buck Bunny.

#### 5.3.1 Transmissão preditiva (vídeo *Coastguard*)

Os resultados apresentados na Tabela 5.12 indicam os valores médios de PSNR e SSIM na transmissão redundante usando método preditivo para o vídeo Coastguard. São indicados os ganhos médios de PSNR obtidos com quarenta simulações usando transmissão sem redundância WIFI e com redundância WIFI + 3G e WIFI + ADSL.

Observa-se que, assim como nos demais métodos de transmissão, os resultados alcançados na transmissão com redundância são melhores quando usado somente um caminho para transmissão do vídeo. Contudo, no método preditivo o ajuste dos

Tabela 5.12: Resultados obtidos para  $\alpha = 0,9$ ,  $\alpha = 1/2$ ,  $\alpha = 1/4$ ,  $\alpha = 1/6$  e  $\alpha = 1/8$  combinados à  $\psi = 1$ ,  $\psi = 3$ ,  $\psi = 5$  e  $\psi = 7$  para vídeo Coastguard.

$\alpha$	Transmissão / $\psi$	PSNR	SSIM	Ganho(dB)	Redundância
$\alpha = 0,9$	WIFI sem redundância	$33,64 \pm 0,97$	$0,881 \pm 0,023$		
	WIFI + 3G	$\psi = 1$ $36,11 \pm 0,35$	$0,939 \pm 0,008$	2,47	15,83%
		$\psi = 3$ $35,93 \pm 0,41$	$0,935 \pm 0,009$	2,29	14,72%
		$\psi = 5$ $36,01 \pm 0,37$	$0,937 \pm 0,008$	2,37	15,19%
		$\psi = 7$ $36,07 \pm 0,36$	$0,938 \pm 0,008$	2,43	15,54%
	WIFI + ADSL	$\psi = 1$ $36,21 \pm 0,27$	$0,941 \pm 0,006$	2,57	15,83%
		$\psi = 3$ $36,04 \pm 0,34$	$0,937 \pm 0,006$	2,40	14,72%
		$\psi = 5$ $36,16 \pm 0,29$	$0,940 \pm 0,006$	2,52	15,19%
		$\psi = 7$ $36,21 \pm 0,27$	$0,941 \pm 0,006$	2,57	15,54%
$\alpha = 1/2$	WIFI + 3G	$\psi = 1$ $36,00 \pm 0,32$	$0,936 \pm 0,007$	2,36	15,56%
		$\psi = 3$ $35,78 \pm 0,40$	$0,932 \pm 0,009$	2,14	13,99%
		$\psi = 5$ $35,94 \pm 0,35$	$0,935 \pm 0,008$	2,30	15,18%
		$\psi = 7$ $35,91 \pm 0,35$	$0,934 \pm 0,008$	2,27	15,05%
	WIFI + ADSL	$\psi = 1$ $36,14 \pm 0,26$	$0,939 \pm 0,006$	2,50	15,56%
		$\psi = 3$ $35,92 \pm 0,35$	$0,935 \pm 0,008$	2,28	13,99%
		$\psi = 5$ $36,07 \pm 0,30$	$0,938 \pm 0,007$	2,43	15,18%
		$\psi = 7$ $36,06 \pm 0,30$	$0,938 \pm 0,006$	2,42	15,05%
$\alpha = 1/4$	WIFI + 3G	$\psi = 1$ $35,70 \pm 0,39$	$0,929 \pm 0,009$	2,06	14,26%
		$\psi = 3$ $35,48 \pm 0,44$	$0,924 \pm 0,010$	1,84	12,49%
		$\psi = 5$ $35,67 \pm 0,40$	$0,929 \pm 0,009$	2,03	14,74%
		$\psi = 7$ $35,64 \pm 0,40$	$0,928 \pm 0,009$	2,00	14,32%
	WIFI + ADSL	$\psi = 1$ $35,82 \pm 0,35$	$0,932 \pm 0,008$	2,18	14,26%
		$\psi = 3$ $35,55 \pm 0,42$	$0,926 \pm 0,009$	1,91	12,49%
		$\psi = 5$ $35,81 \pm 0,37$	$0,932 \pm 0,008$	2,17	14,74%
		$\psi = 7$ $35,78 \pm 0,37$	$0,931 \pm 0,008$	2,14	14,32%
$\alpha = 1/6$	WIFI + 3G	$\psi = 1$ $35,42 \pm 0,47$	$0,923 \pm 0,011$	1,78	12,65%
		$\psi = 3$ $35,17 \pm 0,51$	$0,918 \pm 0,012$	1,53	10,99%
		$\psi = 5$ $35,38 \pm 0,48$	$0,922 \pm 0,011$	1,74	13,31%
		$\psi = 7$ $35,32 \pm 0,48$	$0,921 \pm 0,011$	1,68	12,84%
	WIFI + ADSL	$\psi = 1$ $35,50 \pm 0,46$	$0,924 \pm 0,010$	1,86	12,65%
		$\psi = 3$ $35,20 \pm 0,50$	$0,919 \pm 0,011$	1,56	10,99%
		$\psi = 5$ $35,50 \pm 0,46$	$0,925 \pm 0,010$	1,86	13,31%
		$\psi = 7$ $35,41 \pm 0,47$	$0,923 \pm 0,010$	1,77	12,84%
$\alpha = 1/8$	WIFI + 3G	$\psi = 1$ $35,12 \pm 0,55$	$0,916 \pm 0,013$	1,48	10,94%
		$\psi = 3$ $34,95 \pm 0,57$	$0,911 \pm 0,013$	1,31	9,41%
		$\psi = 5$ $35,15 \pm 0,55$	$0,916 \pm 0,012$	1,51	11,56%
		$\psi = 7$ $35,06 \pm 0,57$	$0,915 \pm 0,013$	1,42	11,10%
	WIFI + ADSL	$\psi = 1$ $35,19 \pm 0,55$	$0,918 \pm 0,012$	1,55	10,94%
		$\psi = 3$ $34,99 \pm 0,56$	$0,913 \pm 0,013$	1,35	9,41%
		$\psi = 5$ $35,26 \pm 0,54$	$0,918 \pm 0,012$	1,62	11,56%
		$\psi = 7$ $35,16 \pm 0,56$	$0,917 \pm 0,013$	1,52	11,10%

parâmetros  $\alpha$  e  $\psi$  resultaram e dados bem interessantes. Pode-se verificar que quanto menor o  $\alpha$  de suavização menores são os ganhos de qualidade, é por exemplo, em 0,9 o maior ganho obtido foi de 2,57 dB, diferente de quando o  $\alpha$  igual a 1/8 com um ganho máximo de 1,62 dB. Esta queda nos ganhos de PSNR também é seguida pelo percentual de redundância de vídeo enviado no caminho secundário. Pode-se observar uma queda de 14,72% com  $\alpha$  igual a 0,9 para menos de 10% com  $\alpha$  igual a 1/8. Esta avaliação confirma o estudo realizado inicialmente quanto ao uso de  $\alpha = 0,9$ .

A avaliação da variável  $\psi$  permite analisar os efeitos de uma maior quantidade de troca de estados. Os resultados apresentados permitem identificar que os menores ganhos ocorrem para  $\psi$  igual a 3, e os maiores ocorrem com  $\psi$  igual a 1 seguido por 5 e 7. O percentual de redundância baseado na constante  $\psi$  também segue uma tendência proporcional ao ganho respectivo, onde quanto maior a redundância maior o ganho. A partir dessa avaliação da qualidade obtida no uso do modelo preditivo para o vídeo Coastguard, é possível aferir que o  $\alpha = 0,9$  e  $\psi = 1$  combinadas resultou em uma degradação menor da qualidade observada.

Comparando os resultados objetivos no envio redundante preditivo com o modelo relativo, observa-se que o preditivo usa menores percentuais de informação redundante enviadas no caminho secundário. A função de mapeamento exponencial calculou uma média de aproximadamente 16,13% de redundância com ganho de qualidade 2,50 dB e 2,64 dB respectivamente utilizando o 3G e ADSL como caminhos secundários. Este percentual reduziu em todos os casos usando predição, contudo essa redução foi seguida na queda dos ganhos de PSNR. Para o melhor resultado de redundância 9,41% obtido com  $\alpha = 1/8$  e  $\psi = 3$  o ganho de PSNR reduziu quase pela metade para 1,31 dB e 1,35 dB, respectivamente para 3G e ADSL. O melhor resultado de ganho de PSNR é identificado para  $\alpha = 0,9$  e  $\psi = 1$  com 2,47 dB e 2,57 dB à um percentual de redundância 15,83% levemente inferior ao método relativo.

O nível de tendência do atraso foi apresentado a partir da variação da constante  $\psi$ . Ao utilizar este método preditivo, contudo considerando apenas o aumento ou queda

real do atraso, são obtidos os resultados apresentando na Tabela 5.13. A máquina de dois estados considera o uso de 0% de redundância com o mínimo de queda do atraso. Os resultados apresentados, indicam uma maior diminuição do percentual de redundância, reduzindo a um mínimo de 6,24% para  $\alpha = 1/8$ . O ganho de qualidade é reduzido para valores de  $\alpha$  menores, o ganho máximo obtido de 1,65 dB é resultado do uso de  $\alpha = 0,9$ . Uma particularidade no uso desta máquina de dois estados, é a proximidade entre os ganhos médios de PSNR entre a utilização do 3G e WIFI como caminhos primários, sendo a diferença máxima observada não sendo superior a 0,08 dB.

Tabela 5.13: Resultados obtidos para  $\alpha = 0,9$ ,  $\alpha = 1/2$ ,  $\alpha = 1/4$ ,  $\alpha = 1/6$  e  $\alpha = 1/8$  combinados à máquina de 2 estados para vídeo Coastguard.

$\alpha$	Transmissão /	PSNR	SSIM	Ganho(dB)	Redundância
$\alpha = 0,9$	WIFI sem redundância	$33,64 \pm 0,97$	$0,881 \pm 0,023$		
	WIFI + 3G	$35,25 \pm 0,53$	$0,919 \pm 0,012$	1,61	10,68%
	WIFI + ADSL	$35,29 \pm 0,51$	$0,920 \pm 0,012$	1,65	10,68%
$\alpha = 1/2$	WIFI + 3G	$34,99 \pm 0,59$	$0,913 \pm 0,013$	1,35	8,86%
	WIFI + ADSL	$35,07 \pm 0,56$	$0,915 \pm 0,013$	1,43	8,86%
$\alpha = 1/4$	WIFI + 3G	$34,79 \pm 0,62$	$0,908 \pm 0,014$	1,15	7,85%
	WIFI + ADSL	$34,87 \pm 0,60$	$0,910 \pm 0,014$	1,23	7,85%
$\alpha = 1/6$	WIFI + 3G	$34,59 \pm 0,66$	$0,904 \pm 0,015$	0,95	6,98%
	WIFI + ADSL	$34,62 \pm 0,66$	$0,905 \pm 0,015$	0,98	6,98%
$\alpha = 1/8$	WIFI + 3G	$34,57 \pm 0,68$	$0,902 \pm 0,016$	0,93	6,24%
	WIFI + ADSL	$34,54 \pm 0,69$	$0,902 \pm 0,016$	0,90	6,24%

### 5.3.2 Transmissão preditiva (vídeo *Big Buck Bunny*)

Os resultados apresentados na Tabela 5.14 para o vídeo Big Buck Bunny indicaram ganhos de qualidade nos níveis obtidos de PSNR e SSIM na transmissão redundante usando método preditivo.

Assim como analisado anteriormente com o vídeo Coastguard, os ganhos de qualidade com o Big Buck Bunny foram significativos quando usado a transmissão redundante com método preditivo. A mesma análise é feita em relação a queda dos valores de  $\alpha$  de 0,9 para 1/8 onde os ganhos em dB do PSNR e percentual de redundância seguiram essa tendência. Seguindo a diminuição dos valores de  $\alpha$ , os percentuais de

Tabela 5.14: Resultados obtidos para  $\alpha = 0,9$ ,  $\alpha = 1/2$ ,  $\alpha = 1/4$ ,  $\alpha = 1/6$  e  $\alpha = 1/8$  combinados à  $\psi = 1$ ,  $\psi = 3$ ,  $\psi = 5$  e  $\psi = 7$  para vídeo Big Buck Bunny.

$\alpha$	Transmissão / $\psi$	PSNR	SSIM	Ganho(dB)	Redundância
$\alpha = 0,9$	WIFI sem redundância	$41,77 \pm 1,27$	$0,950 \pm 0,010$		
	WIFI + 3G	$\psi = 1$ $44,16 \pm 0,75$	$0,973 \pm 0,006$	2,39	10,95%
		$\psi = 3$ $44,17 \pm 0,74$	$0,973 \pm 0,006$	2,40	10,96%
		$\psi = 5$ $44,09 \pm 0,75$	$0,972 \pm 0,006$	2,32	10,61%
		$\psi = 7$ $44,31 \pm 0,60$	$0,974 \pm 0,005$	2,54	11,41%
	WIFI + ADSL	$\psi = 1$ $44,26 \pm 0,73$	$0,974 \pm 0,006$	2,49	10,95%
		$\psi = 3$ $44,29 \pm 0,72$	$0,974 \pm 0,006$	2,52	10,96%
		$\psi = 5$ $44,23 \pm 0,68$	$0,974 \pm 0,006$	2,46	10,61%
		$\psi = 7$ $44,45 \pm 0,55$	$0,976 \pm 0,005$	2,68	11,41%
$\alpha = 1/2$	WIFI + 3G	$\psi = 1$ $43,88 \pm 0,75$	$0,971 \pm 0,006$	2,11	10,26%
		$\psi = 3$ $43,88 \pm 0,75$	$0,971 \pm 0,006$	2,11	10,12%
		$\psi = 5$ $43,75 \pm 0,77$	$0,970 \pm 0,006$	1,98	9,78%
		$\psi = 7$ $44,00 \pm 0,60$	$0,973 \pm 0,005$	2,23	10,52%
	WIFI + ADSL	$\psi = 1$ $43,97 \pm 0,74$	$0,972 \pm 0,006$	2,20	10,26%
		$\psi = 3$ $43,98 \pm 0,75$	$0,972 \pm 0,006$	2,21	10,12%
		$\psi = 5$ $43,89 \pm 0,70$	$0,971 \pm 0,006$	2,12	9,78%
		$\psi = 7$ $44,04 \pm 0,56$	$0,973 \pm 0,005$	2,27	10,52%
$\alpha = 1/4$	WIFI + 3G	$\psi = 1$ $43,27 \pm 0,83$	$0,966 \pm 0,007$	1,50	9,21%
		$\psi = 3$ $43,23 \pm 0,83$	$0,966 \pm 0,007$	1,46	9,08%
		$\psi = 5$ $43,21 \pm 0,82$	$0,966 \pm 0,007$	1,44	8,93%
		$\psi = 7$ $43,33 \pm 0,70$	$0,967 \pm 0,006$	1,56	8,68%
	WIFI + ADSL	$\psi = 1$ $43,32 \pm 0,82$	$0,967 \pm 0,007$	1,55	9,21%
		$\psi = 3$ $43,29 \pm 0,83$	$0,967 \pm 0,007$	1,52	9,08%
		$\psi = 5$ $43,31 \pm 0,77$	$0,967 \pm 0,006$	1,54	8,93%
		$\psi = 7$ $43,41 \pm 0,68$	$0,968 \pm 0,005$	1,64	9,68%
$\alpha = 1/6$	WIFI + 3G	$\psi = 1$ $42,84 \pm 0,89$	$0,964 \pm 0,007$	1,07	8,47%
		$\psi = 3$ $42,79 \pm 0,91$	$0,964 \pm 0,007$	1,02	8,41%
		$\psi = 5$ $42,81 \pm 0,88$	$0,964 \pm 0,007$	1,04	8,44%
		$\psi = 7$ $43,23 \pm 0,72$	$0,966 \pm 0,006$	1,46	9,14%
	WIFI + ADSL	$\psi = 1$ $42,98 \pm 0,88$	$0,965 \pm 0,007$	1,21	8,47%
		$\psi = 3$ $42,89 \pm 0,89$	$0,964 \pm 0,007$	1,12	8,41%
		$\psi = 5$ $42,95 \pm 0,84$	$0,965 \pm 0,006$	1,18	8,44%
		$\psi = 7$ $43,35 \pm 0,69$	$0,968 \pm 0,005$	1,58	9,14%
$\alpha = 1/8$	WIFI + 3G	$\psi = 1$ $42,95 \pm 1,32$	$0,964 \pm 0,007$	1,18	7,69%
		$\psi = 3$ $42,94 \pm 1,33$	$0,964 \pm 0,007$	1,17	7,65%
		$\psi = 5$ $42,90 \pm 1,30$	$0,964 \pm 0,006$	1,13	7,68%
		$\psi = 7$ $43,04 \pm 1,13$	$0,965 \pm 0,006$	1,27	8,48%
	WIFI + ADSL	$\psi = 1$ $42,93 \pm 1,33$	$0,964 \pm 0,007$	1,16	7,69%
		$\psi = 3$ $42,91 \pm 1,35$	$0,964 \pm 0,007$	1,14	7,65%
		$\psi = 5$ $42,96 \pm 1,27$	$0,964 \pm 0,006$	1,19	7,68%
		$\psi = 7$ $43,08 \pm 1,10$	$0,965 \pm 0,006$	1,31	8,48%

redundância também diminuíram de 11,41% para até 7,65% variando de um  $\alpha = 0,9$  para  $\alpha = 1/8$ . Se comparado ao percentual calculado no modelo relativo, onde apresentou um valor de 12,91% houve uma diminuição considerável desse valor. O ganho de qualidade obtido no modelo relativo ainda é maior que o preditivo, sendo 2,96 dB e 3,12 dB contra 2,54 dB e 2,68 dB, respectivamente para 3G e ADSL utilizados como caminho secundário. Diferente do vídeo Coastguard, onde os melhores resultados foram alcançados com  $\psi = 1$ , o Big Buck Bunny apresentou uma melhor qualidade a partir do uso do  $\psi = 7$  para todos os valores de  $\alpha$ .

São apresentados na Tabela 5.15 o resultado com a utilização da máquina de dois estados. Para o vídeo estudado, identifica-se que o ganho diminuiu com o uso de  $\alpha$  menores, quase pela metade comparando  $\alpha$  igual a 0,9 e 1/8. Como o percentual de redundância se aproxima de zero, o ganho de qualidade diminui de 1,97 dB para 0,85 dB com 3G e 2,03 dB para 0,89 dB usando ADSL.

Tabela 5.15: Resultados obtidos para  $\alpha = 0,9$ ,  $\alpha = 1/2$ ,  $\alpha = 1/4$ ,  $\alpha = 1/6$  e  $\alpha = 1/8$  combinados à máquina de 2 estados para vídeo Big Buck Bunny.

$\alpha$	Transmissão /	PSNR	SSIM	Ganho(dB)	Redundância
$\alpha = 0,9$	WIFI sem redundância	$41,77 \pm 1,27$	$0,950 \pm 0,010$		
	WIFI + 3G	$43,74 \pm 0,71$	$0,969 \pm 0,006$	1,97	8,59%
	WIFI + ADSL	$43,80 \pm 0,69$	$0,970 \pm 0,006$	2,03	8,59%
$\alpha = 1/2$	WIFI + 3G	$43,45 \pm 0,74$	$0,965 \pm 0,006$	1,68	6,71%
	WIFI + ADSL	$43,51 \pm 0,73$	$0,966 \pm 0,006$	1,74	6,71%
$\alpha = 1/4$	WIFI + 3G	$43,02 \pm 0,77$	$0,962 \pm 0,007$	1,25	5,82%
	WIFI + ADSL	$43,03 \pm 0,76$	$0,962 \pm 0,007$	1,36	5,82%
$\alpha = 1/6$	WIFI + 3G	$42,80 \pm 0,74$	$0,959 \pm 0,007$	1,03	5,42%
	WIFI + ADSL	$42,85 \pm 0,75$	$0,959 \pm 0,008$	1,08	5,42%
$\alpha = 1/8$	WIFI + 3G	$41,85 \pm 0,75$	$0,955 \pm 0,008$	0,85	4,89%
	WIFI + ADSL	$42,66 \pm 0,75$	$0,956 \pm 0,008$	0,89	4,89%

O modelo preditivo de dois estados apresentou melhores resultado quanto a quantidade de informação redundante, contudo ao comparar com a máquina de cinco estados o seu ganho de qualidade é inferior.

## CAPÍTULO 6

### CONCLUSÕES E TRABALHOS FUTUROS

Um dos grandes avanços nos sistemas de transmissão de conteúdo multimídia é proveniente das melhorias de infraestrutura para melhoria de qualidade. O protocolo SCTP vem sendo utilizado e aplicado em diversos sistemas de transmissão de vídeo como uma alternativa promissora para melhoria de qualidade. O protocolo SCTP possui melhorias em relação aos protocolos convencionais como o *multihoming*. O *multihoming* permite a retransmissão de dados utilizando canais alternativos ou secundários. Uma possibilidade é a troca de caminho mediante a detecção de baixos níveis de qualidade do canal primário.

Neste trabalho foi abordado o uso dos caminhos secundários para envio de pacotes redundantes com o objetivo de melhorar a qualidade de reprodução de vídeo *streaming*.

O uso do *multihoming* SCTP na transmissão dos vídeos com redundância representou melhorias de qualidade baseado nas métricas de avaliação objetiva PSNR e SSIM. Em uma estimativa de transmissão *bicasting* de redundância, ou seja, transmitindo o mesmo vídeo simultaneamente pelo caminho primário e secundário resultou em melhorias de qualidade significativas. O RSTM utiliza caminhos secundários disponíveis para transmitir dados redundantes. A quantidade de informação redundante foi calculada a partir de funções de mapeamento. Os percentuais são obtidos a partir dos atrasos dos caminhos primários. Para todos os vídeos em teste o RSTM levou a uma melhoria das métricas de qualidade. Foram obtidos ganhos quase de mesma ordem da transmissão de 100% de redundância, mas com menor custo de redundância.

Ao aplicar modelos de redundância aliados a um sistema de predição com o objetivo de considerar tendência de aumento ou queda do atraso, foi possível obter ganhos

de qualidade inferiores com um pequeno ganho na redução da redundância.

Foi mostrado que o uso do método RSTM pode prevenir a degradação da qualidade sem gerar sobrecarga excessiva com o uso de recursos desnecessários da rede. Embora os testes tenham utilizado o padrão de codificação H.264, espera-se que bons resultados possam ser obtidos nas demais versões do MPEG ou em outros algoritmos de codificação que exploram as características de redundância temporal para compactação. No método há uma dependência entre as camadas de aplicação e transporte, sendo possível de adequações distribuídas nessas camadas para implementação do método no lado do servidor.

Como sugestão de continuidade deste trabalho, o método proposto pode ser implementado no simulador de redes NS-2 ou NS-3, ou mesmo em um sistema operacional de código aberto. Para implementação em larga escala, é importante realizar estudos de escalabilidade da solução proposta e estudar os efeitos combinados do método proposto e da troca de caminho primário.



## APÊNDICE A

### CÓDIGO FONTE DO GERADOR DE ATRASO REC G.1050

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <string.h>
5  #include "delayGen.h"
6
7  int DEBUG=FALSE;
8  int n=100; //number of samples to generate
9  int carlos=FALSE;
10 int header=TRUE;
11
12 // Severity: A B C D E F G, respectively configured as 0 1 2 3 4 5 6 7
13 int SEVERITY_LANA=1;
14 int SEVERITY_LANB=3;
15 int SEVERITY_ACCESSA=1;
16 int SEVERITY_ACCESSB=6;
17 int SEVERITY_CORE=3;
18
19 // Severity: A B C D E F G
20 // See ITU G.1050 page 23
21 double LAN_occupancy[] = {1, 2, 3, 5, 8, 12, 16, 20};
22 double ACCESS_occupancy[] = {0, 1, 2, 4, 8, 15, 30, 50};
23 double MTU[] = {512, 512, 1508, 1508, 1508, 1508, 1508, 1508};
24 double DELAYCORE[] = {4, 8, 16, 32, 64, 128, 256, 512};
25
26 struct delayctrl dLANA, dACCESSA, dACCESSB, dLANB;
27 struct lossctrl lLANA, lACCESSA, lACCESSB, lLANB;
28
29 double d=0, scale=1000.0;
30
31 void usage(char *prog)
32 {
33     printf("-----");
34     printf("\n Syntetic delay generation according ITU G.1050");
35     printf("\n\tUsage: %s <options>", prog);
36     printf("\n\t<options>:\t-severityLANA:<SEVERITY>\n\t\t\t-severityLANB:<SEVERITY>\n\t\t\t-severityACCESSA:<SEVERITY>\n\t\t\t-severityACCESSB:<SEVERITY>\n\t\t\t-severityCORE:<SEVERITY>\n\t\t\t-samples:<NUMBER>,\n\t\t\t-debug,\n\t\t\t-noheader,\n\t\t\t-help");
37     printf("\n\t\tExample: %s -severityLANA:B -severityLANB:C -severityACCESSA:H -severityACCESSB:G -severityCORE:A -debug -samples:1000", prog);
38
39     printf("\n\n\t\tAttention: \n\t\t\tt1. All options are case sensitive. Default values are ---> LANA:%c ACCESSA:%c LANB:%c ACCESSB:%c CORE:%c", SEVERITY_LANA+'A', SEVERITY_ACCESSA+'A', SEVERITY_LANB+'A', SEVERITY_ACCESSB+'A', SEVERITY_CORE+'A');
40     printf("\n\t\tt2. Core delay is only regional and route flap is not implemented.");
41     printf("\n\t\tt3. Time scale is one millisecond.");
42     printf("\n\n\t\tImplemented by Carlos Marcelo Pedroso");
43     printf("\n\t\tFederal University of Parana - pedroso@eletrica.ufpr.br");
44     printf("\n\t\thttp://www.eletrica.ufpr.br/pedroso\n");
45     printf("\n-----\n\n");
46 }
47
48
49
50 void commandLineParser(int argc, char **argv)
51 {
52     int i;
53     char *s;
54     for (i=1; i<argc; i++) {
55         if ( (s=strstr(argv[i], "-severityLANA")) )
56         {
57             s=strstr(argv[i], ":");
58             SEVERITY_LANA = *(s+1)-'A';
59             if ( (SEVERITY_LANA<0)|| (SEVERITY_LANA>7) )
60             {
61                 printf("\n\n Error: SEVERITY_LANA must be choosen between A and H. See ITU G.1050 page 23\n\n");
62                 usage(argv[0]);
63                 exit(0);
64             }
65             continue;
66         }
67         if ( (s=strstr(argv[i], "-severityLANB")) )
68         {
69             s=strstr(argv[i], ":");
70             SEVERITY_LANB = *(s+1)-'A';
71             if ( (SEVERITY_LANB<0)|| (SEVERITY_LANB>7) )
72             {
73                 printf("\n\n Error: SEVERITY_LANB must be choosen between A and H. See ITU G.1050 page 23\n\n");
74                 usage(argv[0]);
75                 exit(0);
76             }
77             continue;
78         }
79         if ( (s=strstr(argv[i], "-severityACCESSA")) )
80         {

```

```

81         s=strstr(argv[i],":");
82         SEVERITY_ACCESSA = *(s+1)-'A';
83         if ( (SEVERITY_ACCESSA<0)|| (SEVERITY_ACCESSA>7) )
84         {
85             printf("\n\n Error: SEVERITY_ACCESSA must be choosen between A and H. See ITU G.1050 page 23\n\n");
86             usage(argv[0]);
87             exit(0);
88         }
89         continue;
90     }
91     if ( (s=strstr(argv[i],"-severityACCESSB" )) )
92     {
93         s=strstr(argv[i],":");
94         SEVERITY_ACCESSB = *(s+1)-'A';
95         if ( (SEVERITY_ACCESSB<0)|| (SEVERITY_ACCESSB>7) )
96         {
97             printf("\n\n Error: SEVERITY_ACCESSB must be choosen between A and H. See ITU G.1050 page 23\n\n");
98             usage(argv[0]);
99             exit(0);
100         }
101         continue;
102     }
103     if ( (s=strstr(argv[i],"-severityCORE" )) )
104     {
105         s=strstr(argv[i],":");
106         SEVERITY_CORE = *(s+1)-'A';
107         if ( (SEVERITY_CORE<0)|| (SEVERITY_CORE>7) )
108         {
109             printf("\n\n Error: SEVERITY_CORE must be choosen between A and H. See ITU G.1050 page 23\n\n");
110             usage(argv[0]);
111             exit(0);
112         }
113         continue;
114     }
115     if ( (s=strstr(argv[i],"-carlos" )) )
116     {
117         carlos=TRUE;
118         continue;
119     }
120     if ( (s=strstr(argv[i],"-noheader" )) )
121     {
122         header=FALSE;
123         continue;
124     }
125     if ( (s=strstr(argv[i],"-debug" )) )
126     {
127         DEBUG=TRUE;
128         continue;
129     }
130     if ( (s=strstr(argv[i],"-help" )) )
131     {
132         usage(argv[0]);
133         exit(0);
134     }
135     if ( (s=strstr(argv[i],"-samples" )) )
136     {
137         s=strstr(argv[i],":");
138         if (!validNumber(s+1)) {
139             printf("\n\n You must specify a valid number of samples\n\n");
140             usage(argv[0]);
141             exit(0);
142         }
143         n=atoi(s+1);
144         continue;
145     }
146
147     if (header)
148         printf("Config to LANA:%c ACCESSA:%c LANB:%c ACCESSB:%c CORE:%c samples:%d", SEVERITY_LANA+'A',
149             SEVERITY_ACCESSA+'A',SEVERITY_LANB+'A',SEVERITY_ACCESSB+'A',SEVERITY_CORE+'A',n);
150
151 }
152
153 void initloss ()
154 {
155     // -----
156     // page 32 II.6.1 derived parameters 1)
157     ILANA.loss_probability[LOW_LOSS]=0;
158     ILANA.loss_probability[HIGH_LOSS]=0.000025 * LAN_occupancy[SEVERITY_LANA];
159     ILANB.loss_probability[LOW_LOSS]=0;
160     ILANB.loss_probability[HIGH_LOSS]=0.000025 * LAN_occupancy[SEVERITY_LANB];
161     // -----
162     // page 32 II.6.1 derived parameters 2)
163     ILANA.transition_probability[LOW_LOSS]=0.0001 * LAN_occupancy[SEVERITY_LANA];
164     ILANA.transition_probability[HIGH_LOSS]=0.1;
165     ILANB.transition_probability[LOW_LOSS]=0.0001 * LAN_occupancy[SEVERITY_LANB];
166     ILANB.transition_probability[HIGH_LOSS]=0.1;
167     // -----
168     // page 33 II.6.2 8)
169     IACCESSA.loss_probability[LOW_LOSS]=0;
170     IACCESSA.loss_probability[HIGH_LOSS]=0.0005 * ACCESS_occupancy[SEVERITY_ACCESSA];
171     IACCESSB.loss_probability[LOW_LOSS]=0;
172     IACCESSB.loss_probability[HIGH_LOSS]=0.0005 * ACCESS_occupancy[SEVERITY_ACCESSB];
173     // -----
174     // page 33 II.6.2 4)
175     IACCESSA.transition_probability[LOW_LOSS]=0.0002 * ACCESS_occupancy[SEVERITY_LANA];
176     IACCESSA.transition_probability[HIGH_LOSS]=0.2/(1+ACCESS_occupancy[SEVERITY_LANA]);
177     IACCESSB.transition_probability[LOW_LOSS]=0.0002 * ACCESS_occupancy[SEVERITY_LANB];
178     IACCESSB.transition_probability[HIGH_LOSS]=0.2/(1+ACCESS_occupancy[SEVERITY_LANB]);

```

```

178
179 // -----
180 // initial loss state is LOW_LOSS
181 ILANA_LOSS.STATE=ILANB_LOSS.STATE=IACCESSA_LOSS.STATE=IACCESSB_LOSS.STATE=LOW_LOSS;
182 ILANA_loss=ILANB_loss=IACCESSA_loss=IACCESSB_loss=FALSE;
183
184 // -----
185 // severity; only to simplify ... sorry
186 ILANA_severity= dLANA_severity= SEVERITY_LANA;
187 ILANB_severity= dLANB_severity=SEVERITY_LANB;
188 IACCESSA_severity= dACCESSA_severity=SEVERITY_ACCESSA;
189 IACCESSB_severity= dACCESSB_severity=SEVERITY_ACCESSB;
190
191 // -----
192 // some initial values for delay (10ms)
193 dLANA_d=dLANB_d=dACCESSA_d=dACCESSB_d=0.01;
194
195 // some config for rate .. see scenario rate combination
196
197 dACCESSA_rate=1000000;
198 dACCESSB_rate=1000000;
199 dLANA_rate=40000000;
200 dLANB_rate=40000000;
201 }
202
203 int validNumber(char *s)
204 {
205     int i;
206     for (i=0;i<strlen(s);i++)
207         if (!isdigit(*(s+i)))
208             return 0;
209
210     return 1;
211 }
212
213 // ITU G.1050 page 30
214 void gilbert_elliott(struct lossctrl *lossptr)
215 {
216     if (drand48() < lossptr->loss_probability[lossptr->LOSS.STATE])
217         lossptr->loss = TRUE;
218     else
219         lossptr->loss = FALSE;
220
221     if (drand48() < lossptr->transition_probability[lossptr->LOSS.STATE])
222     if (lossptr->LOSS.STATE == HIGH_LOSS)
223         lossptr->LOSS.STATE = LOW_LOSS;
224     else
225         lossptr->LOSS.STATE = HIGH_LOSS;
226 }
227
228 // ITU G.1050 page 31
229 void filter(double impulse_probability, double impulse_height, double A, double *d)
230 {
231     double i;
232
233     if (drand48() < impulse_probability)
234         i = impulse_height;
235     else
236         i = 0;
237
238     // G.1050 define the filter using two different equations.
239     // item 11.3 define as  $d(n) = d(n-1) * (A) + (\text{impulse height}) * (1-A)$ 
240     // and after 11.6.1 5) and 11.6.2 7), in the algorithm for parameter setting, as  $dn = A \hat{A} ? (\text{impulse height}) + (1 \hat{A} ? ? A$ 
241     //  $\hat{A} ? (\text{previous delay})$ 
242     // tests indicate that probably the second is correct...
243     (*d) = i*(A) + (*d) * (1-A);
244 }
245
246 // ITU G.1050 page 32
247 void filterLAN(struct delayctrl *delayptr, struct lossctrl *lossptr)
248 {
249     // -----
250     // page 32 11.6.1 subitem 3, 4, and 5
251     if (lossptr->loss==LOW_LOSS) {
252         delayptr->impulse_height=0;
253         delayptr->impulse_probability=0;
254         delayptr->A=1;
255     }
256     else {
257         delayptr->impulse_height= drand48() * ( (MTU[delayptr->severity]*8.0/delayptr->rate) * (1.0 + LAN_occupancy[
258             delayptr->severity]/40.0) );
259         delayptr->impulse_probability=0.5;
260         delayptr->A=1;
261     }
262
263     filter(delayptr->impulse_probability, delayptr->impulse_height, delayptr->A, &delayptr->d);
264     //delayptr->d += MTU[delayptr->severity]/delayptr->rate; // assume all packets with fixed size equal to link MTU
265 }
266
267 // ITU G.1050 page 33
268 void filterACCESS(struct delayctrl *delayptr, struct lossctrl *lossptr)
269 {
270     // -----
271     // page 32 11.6.2 subitem 5,
272     // scale factor: see 11.3.1
273     if (lossptr->loss==LOW_LOSS) {
274         delayptr->impulse_height= scale * drand48() * ( 0.25 * (MTU[delayptr->severity]*8.0/delayptr->rate) * (1.0 +

```

```

274         ACCESS.occupancy[delayptr->severity]/40.0));
275         delayptr->impulse_probability= 0.001 + ACCESS.occupancy[delayptr->severity]/2000.0;
276         delayptr->A=0.25;
277     }
278     else {
279         delayptr->impulse_height= scale * 0.25 * (MTU[delayptr->severity]*8.0/delayptr->rate) * (1.0 +
280             ACCESS.occupancy[delayptr->severity]/40.0);
281         delayptr->impulse_probability=0.3+ 0.4 * ACCESS.occupancy[delayptr->severity]/100.0;
282         delayptr->A=0.25;
283     }
284     filter(delayptr->impulse_probability, delayptr->impulse_height, delayptr->A, &delayptr->d);
285     //delayptr->d += MTU[delayptr->severity]/delayptr->rate; // assume all packets with fixed size equal to link MTU
286 }
287 double dCORE()
288 {
289     return ( (drand48() * DELAYCORE[SEVERITY.CORE])/1000) ;
290 }
291
292 void main(int argc, char **argv)
293 {
294     double u, d=100, x, soma=0, j1=0, j2=0;
295     int i;
296
297     CommandLineParser(argc, argv);
298
299     srand48(time(NULL));
300     initloss();
301     for (i=0,d=0.1;i<n;i++)
302     {
303         gilbert_elliott(&ILANA);
304         gilbert_elliott(&ILANB);
305         gilbert_elliott(&IACCESSA);
306         gilbert_elliott(&IACCESSB);
307
308         filterLAN (&dLANA,&ILANA);
309         filterLAN (&dLANB,&ILANB);
310         filterACCESS (&dACCESSA,&IACCESSA);
311         filterACCESS (&dACCESSB,&IACCESSB);
312
313         d= dLANA.d + MTU[dLANA.severity]*8.0/dLANA.rate + dACCESSA.d + MTU[dACCESSA.severity]*8.0/dACCESSA.rate + dLANB.d +
            MTU[dLANB.severity]*8.0/dLANB.rate+ dACCESSB.d + dACCESSA.d + MTU[dACCESSB.severity]*8.0/dACCESSB.rate + dCORE()
            ;
314
315
316         if (DEBUG) {
317             printf("\n -----");
318             printf("\n Delay");
319             printf("\n LANA=%f ACCESSA=%f LANB=%f ACCESSB=%f TOTAL=%f",dLANA.d,dACCESSA.d ,dLANB.d ,dACCESSB.d,d);
320             printf("\n Loss");
321             printf("\n LossState: LANA=%s ACCESSA=%s LANB=%s ACCESSB=%s",ILANA.LOSS.STATE=="0" ? "LOW_LOSS": "HIGH_LOSS",
322                 ILANB.LOSS.STATE=="0" ? "LOW_LOSS": "HIGH_LOSS", IACCESSA.LOSS.STATE=="0" ? "LOW_LOSS": "HIGH_LOSS", IACCESSB.
323                 LOSS.STATE=="0" ? "LOW_LOSS": "HIGH_LOSS");
324             printf("\n Loss: LANA=%d ACCESSA=%d LANB=%d ACCESSB=%d",ILANA.loss,IACCESSA.loss,ILANB.loss,IACCESSB.loss);
325             printf("\n Probability: LANA[%f, %f] ACCESSA[%f, %f] LANB[%f, %f] ACCESSB[%f, %f]", ILANA.loss_probability
326                 [0],ILANA.loss_probability[1], IACCESSA.loss_probability[0],IACCESSA.loss_probability[1], ILANB.
327                 loss_probability[0],ILANB.loss_probability[1], IACCESSB.loss_probability[0],IACCESSB.loss_probability
328                 [1]);
329             printf("\n Transition: LANA[%f, %f] ACCESSA[%f, %f] LANB[%f, %f] ACCESSB[%f, %f]", ILANA.
330                 transition_probability[0],ILANA.transition_probability[1], IACCESSA.transition_probability[0],IACCESSA.
331                 transition_probability[1], ILANB.transition_probability[0],ILANB.transition_probability[1], IACCESSB.
332                 transition_probability[0],IACCESSB.transition_probability[1]);
333         }
334     }
335     else
336         /* for carlos gouvea: average delay for one second */
337         if (carlos) {
338             soma += d;
339             if (i%1000==0) {
340                 printf("\n%f", soma/1000);
341                 soma=0;
342             }
343         }
344     }
345     else
346         printf("\n%f",d);
347 }
348
349     printf("\n");
350 }

```

# APÊNDICE B

## SCRIPT EM BASH

```

1  #Parâmetros de entrada
2  video=$1          #Nome do Vídeo
3  primario=$2       #Indicação do caminho primário
4  secundario=$3     #Indicação do caminho secundário
5  metodo=$4        #Método de redundância relativo
6  alpha=$5         #alpha suavizador
7  teta=$6          #angulo preditivo
8
9  #Cria o video no formato m4v
10 rm $video.m4v
11 ffmpeg -s cif -r 25 -f rawvideo -vcodec rawvideo -pix_fmt yuv420p -i $video'_.cif.yuv' -vcodec libx264 -g 12 -b_strategy 0 -bf
12 2 $video.m4v
13 #O parâmetro de resolução anterior é alterado para cada tipo de formato de video QCIF, CIF e HD
14 #Coloca o m4v no container mp4
15 rm $video.mp4
16 MP4Box -hint -mtu 1460 -fps 25 -add $video.m4v $video.mp4
17
18 #gera a lista de frames enviados
19 rm st.$video.txt
20 mp4trace -f -s 200.17.220.1 12346 $video.mp4 > st.$video.txt
21
22 #gera a lista de pacotes enviados
23 rm st.$video-packet.txt
24 mp4trace -p -s 200.17.220.1 12346 $video.mp4 > st.$video-packet.txt
25
26 #Formata a lista de pacotes para poder ser trabalhado
27 rm sent-$video.cap
28 cat st.$video-packet.txt | awk '{ print $3"\t", "id \"$1"\t", "udp", "\t"$4, "\t"$2}' > sent-$video.cap
29
30 rm sent1-$video.cap
31 cat st.$video-packet.txt | awk '{ print $3"\t", "id \"$1"\t", "udp", "\t"$4}' > sent1-$video.cap
32
33 #-----Calcula redundância a ser enviada-----
34 rm redundancia_exp.txt
35 rm redundancia_2grau.txt
36 rm redundancia_linear.txt
37 rm suavizacao.txt
38 rm wifi_ufpr.cap
39 rm wifi_ufpr1.txt
40
41 cat $primario.txt | awk '{ print $1 * 1000}' > $primario_1.txt #Converte tempos de ms em segundos
42 cat $primario_1.txt | awk '{ print $1}' > $primario.cap
43 valor=$(wc -l < $primario.cap) #le a quantidade de linhas do arquivo
44 x=( $(cat $primario_1.txt) )
45 suavizado=( 0 )
46 lambda=0.04
47 tau=0
48 j=0
49
50 while [ $j -lt $valor ]
51 do
52     if [ $j -le 0 ] #diferencia a primeira linha, logica diferente
53     then
54         suavizado[j]=$x[j]
55         echo ${suavizado[j]} >>> suavizacao.txt
56     else
57         suavizado[j]=$(bc <<< "scale=5; (1 - $alpha) * ${suavizado[j-1]} + $alpha * ${x[j]}")
58         echo ${suavizado[j]} >>> suavizacao.txt
59     fi
60
61     dif1=$(bc <<< "200 - ${suavizado[j]}")
62     dif1=${dif1%.*}
63
64     if [ "$dif1" -ge 0 ]
65     then
66         diferenca[j]=$(bc <<< "scale=6; (200 - ${suavizado[j]} / 2)")
67     else
68         diferenca[j]=0
69     fi
70
71     dif2=${diferenca[j]%.*} #converte valor float para int a ser usado na comparação if
72
73     if [ $dif2 -le $tau ]
74     then
75         redundancia2grau[j]=1
76         redundancia1grau[j]=1
77         redundanciaexp[j]=1
78         echo ${redundanciaexp[j]} >>> redundancia_exp.txt
79         echo ${redundancia2grau[j]} >>> redundancia_2grau.txt
80         echo ${redundancia1grau[j]} >>> redundancia_linear.txt
81     else
82         redundanciaexp[j]=$(echo "scale=4; e( - $lambda * ${diferenca[j]} )" | bc -l)
83         redundancia2grau[j]=$(bc <<< "scale=4; (( 1 / ( 200 - $tau )) * (( 200 - $tau ) - ( ${diferenca[j]} - $tau)))
            ^ 2")

```

```

84         redundanciaLinear[j]=$((bc <<< "scale=4; (( 1 / ( 200 - $tau )) * (( 200 - $tau ) - ( ${diferenca[j]} - $tau)))
85         echo ${redundanciaexp[j]} >> redundancia_exp.txt
86         echo ${redundancia2grau[j]} >> redundancia_2grau.txt
87         echo ${redundanciaLinear[j]} >> redundancia_linear.txt
88     fi
89     j=$(( $j + 1 ))
90 done
91 #-----Calculo redundância preditiva a ser enviada-----
92 atah()
93 {
94     echo "scale=3;a($1)/0.017453293" | bc -l          #Calcula o angulo de \theta
95 }
96
97 j=0
98 rm redundancia_nova.$metodo.txt
99 x=( ${cat $primario.txt} )                          #Arquivo com os tempos de atraso canal primário
100 y=( ${cat redundancia.$metodo.txt} )                 #Percentuais de redundância método relativo
101
102 estadoatual=10
103 while [ $j -lt $valor ] #loop funciona enquanto
104 do
105     if [ $j -eq 0 ]
106     then
107         echo ${y[j]} >> redundancia_nova.$metodo.txt
108     else
109         atrasodiferenca=$((bc <<< "${x[j]} - ${x[j-1]}" ))
110         angulo=$(atah $atrasodiferenca)
111         if [ $estadoatual == 20 ]
112         then
113             if [ "$( echo $angulo '>=' -$teta | bc -l )" -eq 1 ]
114             then
115                 novaredundante=$((bc <<< "${y[j]} * 2 ")
116                 echo $novaredundante >> redundancia_nova.$metodo.txt
117                 estadoatual=20
118             else
119                 novaredundante=$((bc <<< "${y[j]} * 1.5 ")
120                 echo $novaredundante >> redundancia_nova.$metodo.txt
121                 estadoatual=15
122             fi
123         else
124             if [ $estadoatual == 15 ]
125             then
126                 if [ "$( echo $angulo '>=' -$teta | bc -l )" -eq 1 ]
127                 then
128                     novaredundante=$((bc <<< "${y[j]} * 2 ")
129                     echo $novaredundante >> redundancia_nova.$metodo.txt
130                     estadoatual=20
131                 else
132                     if [ "$( echo $angulo '<=' -$teta | bc -l )" -eq 1 ]
133                     then
134                         novaredundante=$((bc <<< "${y[j]} * 1 ")
135                         echo $novaredundante >> redundancia_nova.$metodo.txt
136                         estadoatual=10
137                     else
138                         novaredundante=$((bc <<< "${y[j]} * 1.5 ")
139                         echo $novaredundante >> redundancia_nova.$metodo.txt
140                         estadoatual=15
141                     fi
142                 fi
143             else
144                 if [ $estadoatual == 10 ]
145                 then
146                     if [ "$( echo $angulo '>=' -$teta | bc -l )" -eq 1 ]
147                     then
148                         novaredundante=$((bc <<< "${y[j]} * 1.5 ")
149                         echo $novaredundante >> redundancia_nova.$metodo.txt
150                         estadoatual=15
151                     else
152                         if [ "$( echo $angulo '<=' -$teta | bc -l )" -eq 1 ]
153                         then
154                             novaredundante=$((bc <<< "${y[j]} * 0.5 ")
155                             echo $novaredundante >> redundancia_nova.$metodo.txt
156                             estadoatual=5
157                         else
158                             novaredundante=$((bc <<< "${y[j]} * 1 ")
159                             echo $novaredundante >> redundancia_nova.$metodo.txt
160                             estadoatual=10
161                         fi
162                     fi
163                 else
164                     if [ $estadoatual == 5 ]
165                     then
166                         if [ "$( echo $angulo '>=' -$teta | bc -l )" -eq 1 ]
167                         then
168                             novaredundante=$((bc <<< "${y[j]} * 1 ")
169                             echo $novaredundante >> redundancia_nova.$metodo.txt
170                             estadoatual=10
171                         else
172                             if [ "$( echo $angulo '<=' -$teta | bc -l )" -eq 1 ]
173                             then
174                                 novaredundante=$((bc <<< "${y[j]} * 0 ")
175                                 echo $novaredundante >> redundancia_nova.$metodo.txt
176                                 estadoatual=0
177                             else
178                                 novaredundante=$((bc <<< "${y[j]} * 0.5 ")
179                                 echo $novaredundante >> redundancia_nova.$metodo.txt
180                                 estadoatual=5

```

```

181                                     fi
182                                     fi
183                                     else
184                                     if [ $estadoatual == 0 ]
185                                     then
186                                         if [ "$( echo $angulo '>=' $teta | bc -l )" -eq 1 ]
187                                         then
188                                             novaredundante=$(bc <<< " ${y[j]} * 0.5 ")
189                                             echo $novaredundante >> redundancia_nova.$metodo.txt
190                                             estadoatual=5
191                                         else
192                                             novaredundante=$(bc <<< " ${y[j]} * 0 ")
193                                             echo $novaredundante >> redundancia_nova.$metodo.txt
194                                             estadoatual=0
195                                         fi
196                                     fi
197                                     fi
198                                     fi
199                                     fi
200                                     fi
201                                     fi
202                                     j=$(( $j + 1 ))
203     done
204
205     rm temporario.txt
206     cat redundancia_nova.$metodo.txt > temporario.txt
207     rm redundancia_nova.$metodo.txt
208     temp=( $(cat temporario.txt) )
209
210     valor=$(wc -l < temporario.txt)
211     w=0
212     while [ $w -lt $valor ] #loop funciona enquanto
213     do
214         if [ "$( echo ${temp[w]} '>' 1 | bc -l )" -eq 1 ] #Percentuais de redundância acima de 100% são mantidos
215         then com 100%
216             echo 1 >> redundancia_nova.$metodo.txt
217         else
218             echo ${temp[w]} >> redundancia_nova.$metodo.txt
219         fi
220         w=$(( $w + 1 ))
221     done
222
223     #-----Linearizador de atrasos para 100 Mbps-----
224     rm sequencia.txt
225     cat st.$video-packet.txt | sort -b -n -k3 -k1 > sequencia.txt
226
227     limite=$(wc -l < sequencia.txt)
228     id=( $(awk '{ print $1}' ./sequencia.txt) ) #identificador do pacote
229     tipo=( $(awk '{ print $2}' ./sequencia.txt) ) #tipo do quadro (I, P, B) do pacote
230     tempo=( $(awk '{ print $3}' ./sequencia.txt) ) #tempo de geração do pacote
231     tamanho=( $(awk '{ print $4}' ./sequencia.txt) ) #tamanho em bytes do pacote
232     t=0
233     taxa=0.00000008 #taxa de transmissão a 100M bps
234
235     rm a1.txt
236     rm a2.txt
237     rm a3.txt
238     rm a4.txt
239
240     while [ $t -lt $limite ]
241     do
242         if [ $t = 0 ]
243         then
244             echo ${id[t]} >> a1.txt
245             echo ${tipo[t]} >> a2.txt
246             echo ${tempo[t]} >> a3.txt
247             anterior_tempo=${tempo[t]}
248             echo ${tamanho[t]} >> a4.txt
249         else
250             if [ "$( echo ${tempo[t]} '>' ${tempo[t-1]} | bc -l )" -eq 1 ]
251             then
252                 echo ${id[t]} >> a1.txt
253                 echo ${tipo[t]} >> a2.txt
254                 echo ${tempo[t]} >> a3.txt
255                 anterior_tempo=${tempo[t]}
256                 echo ${tamanho[t]} >> a4.txt
257             else
258                 atual_tempo=$(bc <<< " $anterior_tempo + ${tamanho[t-1]} * $taxa ")
259                 echo ${id[t]} >> a1.txt
260                 echo ${tipo[t]} >> a2.txt
261                 echo $atual_tempo | bc | awk '{ printf "%f\n", $0}' >> a3.txt
262                 anterior_tempo=$atual_tempo
263                 echo ${tamanho[t]} >> a4.txt
264             fi
265         fi
266         t=$(( $t + 1 ))
267     done
268
269     rm tempos_transmissao.txt
270     paste a1.txt a2.txt a3.txt a4.txt >> tempos_transmissao.txt
271
272     limite=$(wc -l < tempos_transmissao.txt)
273     tempo=( $(awk '{ print $3}' ./tempos_transmissao.txt) ) #tempos base de criação dos pacotes
274     t=0
275     z=1
276     cont=0
277

```

```

278 rm contador.txt
279 while [ $t -lt $limite ]
280 do
281     if [ "$( echo ${tempo[t]} ' <' $z | bc -l )" -eq 1 ]
282     then
283         cont=$(( $cont + 1 ))
284     else
285         echo $cont >> contador.txt
286         z=$(( $z + 1 ))
287         cont=1
288     fi
289     t=$(( $t + 1 ))
290 done
291 echo $cont >> contador.txt
292
293 rm new_tempo_3g.txt
294 rm new_tempo_wifi.txt
295 rm new_tempo_adsl.txt
296
297 limite=$((wc -l < tempos_transmissao.txt))
298 tempo=( $(awk '{print $3}' ./tempos_transmissao.txt) ) #tempos base de criação dos pacotes
299 quantidade=( $(awk '{print $1}' ./contador.txt) ) #quantidade de pacotes base de criação
300 atraso_3g=( $(awk '{print $1}' ./3g.txt) ) #tempos 3G base de criação dos pacotes
301 atraso_wifi=( $(awk '{print $1}' ./wifi-ufpr.txt) ) #tempos WIFI base de criação dos pacotes
302 atraso_adsl=( $(awk '{print $1}' ./adsl.casa.txt) ) #tempos ADSL base de criação dos pacotes
303 t=0
304 z=1
305 cont=0
306
307 while [ $t -lt $limite ]
308 do
309     if [ "$( echo ${tempo[t]} ' <' $z | bc -l )" -eq 1 ]
310     then
311         diferenca_3g=$((bc <<< " scale=10; ( ${atraso_3g[z]} - ${atraso_3g[z-1]} ) / ${quantidade[z-1]} " ))
312         diferenca_wifi=$((bc <<< " scale=10; ( ${atraso_wifi[z]} - ${atraso_wifi[z-1]} ) / ${quantidade[z-1]} " ))
313         diferenca_adsl=$((bc <<< " scale=10; ( ${atraso_adsl[z]} - ${atraso_adsl[z-1]} ) / ${quantidade[z-1]} " ))
314
315         new_tempo_3g=$((bc <<< " ${tempo[t]} + ${atraso_3g[z-1]} + $diferenca_3g * $cont " ))
316         new_tempo_wifi=$((bc <<< " ${tempo[t]} + ${atraso_wifi[z-1]} + $diferenca_wifi * $cont " ))
317         new_tempo_adsl=$((bc <<< " ${tempo[t]} + ${atraso_adsl[z-1]} + $diferenca_adsl * $cont " ))
318
319         echo $new_tempo_3g | bc | awk '{printf "%f\n", $0}' >> new_tempo_3g.txt
320         echo $new_tempo_wifi | bc | awk '{printf "%f\n", $0}' >> new_tempo_wifi.txt
321         echo $new_tempo_adsl | bc | awk '{printf "%f\n", $0}' >> new_tempo_adsl.txt
322
323         cont=$(( $cont + 1 ))
324     else
325         new_tempo_3g=$((bc <<< " ${tempo[t]} + ${atraso_3g[z]} " ))
326         new_tempo_wifi=$((bc <<< " ${tempo[t]} + ${atraso_wifi[z]} " ))
327         new_tempo_adsl=$((bc <<< " ${tempo[t]} + ${atraso_adsl[z]} " ))
328
329         echo $new_tempo_3g | bc | awk '{printf "%f\n", $0}' >> new_tempo_3g.txt
330         echo $new_tempo_wifi | bc | awk '{printf "%f\n", $0}' >> new_tempo_wifi.txt
331         echo $new_tempo_adsl | bc | awk '{printf "%f\n", $0}' >> new_tempo_adsl.txt
332
333         z=$(( $z + 1 ))
334         cont=1
335     fi
336     t=$(( $t + 1 ))
337 done
338
339 rm arquivotemp1.txt
340 rm arquivotemp2.txt
341 rm arquivotemp3.txt
342
343 rm recv-$video-3g.cap
344 rm recv-$video-wifi.cap
345 rm recv-$video-adsl.cap
346
347 rm sent-redundante-$video-3g.cap
348 rm sent-redundante-$video-wifi.cap
349 rm sent-redundante-$video-adsl.cap
350
351 paste tempos_transmissao.txt new_tempo_3g.txt >> arquivotemp1.txt
352 cat arquivotemp1.txt | awk '{print $5"\t", "id \"$1\"\t","udp","\t"$4}' > recv-$video-3g.cap
353 cat arquivotemp1.txt | awk '{print $5"\t", "id \"$1\"\t","udp","\t"$4,"\t"$2}' > sent-redundante-$video-3g.cap
354
355 paste tempos_transmissao.txt new_tempo_wifi.txt >> arquivotemp2.txt
356 cat arquivotemp2.txt | awk '{print $5"\t", "id \"$1\"\t","udp","\t"$4}' > recv-$video-wifi.cap
357 cat arquivotemp2.txt | awk '{print $5"\t", "id \"$1\"\t","udp","\t"$4,"\t"$2}' > sent-redundante-$video-wifi.cap
358
359 paste tempos_transmissao.txt new_tempo_adsl.txt >> arquivotemp3.txt
360 cat arquivotemp3.txt | awk '{print $5"\t", "id \"$1\"\t","udp","\t"$4}' > recv-$video-adsl.cap
361 cat arquivotemp3.txt | awk '{print $5"\t", "id \"$1\"\t","udp","\t"$4,"\t"$2}' > sent-redundante-$video-adsl.cap
362
363 rm temp1.txt
364 rm temp2.txt
365 rm temp3.txt
366 cat recv-$video-3g.cap | sort -b -n -k3 -k1 > temp1.txt
367 cat recv-$video-wifi.cap | sort -b -n -k3 -k1 > temp2.txt
368 cat recv-$video-adsl.cap | sort -b -n -k3 -k1 > temp3.txt
369
370 rm recv-$video-3g.cap
371 rm recv-$video-wifi.cap
372 rm recv-$video-adsl.cap
373
374 cat temp1.txt > recv-$video-3g.cap
375 cat temp2.txt > recv-$video-wifi.cap

```



```

376 cat temp3.txt > recv-$video-ads1.cap
377
378 #-----Gerador de arquivos com os pacotes redundantes enviados, parte 1-----
379
380 #Gerador arquivo redundantes
381 rm recv.redundante-$video-$secundario.cap
382 rm framel.txt
383 rm melhor.txt
384 cat sent-redundante-$video-$secundario.cap | sort -b -n -k1 -k3 > melhor.txt
385 rm sent-$video.cap
386 cat melhor.txt > sent-$video.cap
387 tamanho=( $(awk '{ print $5}' ./sent-$video.cap) )      #tamanhos dos pacotes
388 tempo=( $(awk '{ print $1}' ./sent-$video.cap) )          #tempos base dos pacotes
389
390 tam_intervalo=0
391 linha=$(wc -l < sent-$video.cap)      #quantidade de linhas arquivo sent
392 t=0
393 x=1
394 rm tamanho.txt
395
396 while [ $t -le $linha ]
397 do
398     dif3=${tempo[t]%.*} #converte valor float para int a ser usado na comparação if
399     if [ $dif3 -lt $x ]
400     then
401         tam_intervalo=$(( $tam_intervalo + ${tamanho[t]} ))
402     else
403         echo $tam_intervalo >> tamanho.txt
404         x=$(( $x + 1 ))
405         tam_intervalo=${tamanho[t]}
406     fi
407     t=$(( $t + 1 ))
408 done
409 redundancia=( $(awk '{ print $1}' ./redundancia.nova.$metodo.txt) )
410 tamanhoseg=( $(awk '{ print $1}' ./tamanho.txt) ) #vetor com o tamanho de cada segundo de video
411
412 k=0
413 soma=0
414
415 rm redundante.txt
416 cat sent-$video.cap | while read line
417 do
418     tempopacote='echo $line | awk '{ print $1 }''
419     tampacote='echo $line | awk '{ print $5 }''
420     tipopacote='echo $line | awk '{ print $6 }''
421     linha='echo $line | awk '{ print $1"\t",$2 "3"\t",$4,"\t"$5 }''
422
423     anterior=$intervalo
424     intervalo=${tempopacote%.*} #converte valor float para int a ser usado na comparação if
425     if [ $anterior -ne $intervalo ]
426     then
427         soma=0
428     fi
429     bytes=$(( bc <<< " (${redundancia[intervalo]} * ${tamanhoseg[intervalo]})" )
430     bytes=${bytes%.*}
431     if [ "$( echo ${redundancia[intervalo]} '>' 0 | bc -l )" -eq 1 ]
432     then
433         if [ "$tipopacote" = "H" -o "$tipopacote" = "I" ]
434         then
435             soma=$(( $soma + $tampacote ))
436
437             if [ "$( echo $soma '<=' $bytes | bc -l )" -eq 1 ]
438             then
439                 echo $linha >> redundante.txt
440             fi
441         fi
442     fi
443 fi
444 done
445
446 j=1
447 soma=0
448 rm teste.txt
449 cat redundante.txt | while read line
450 do
451     tempopacote='echo $line | awk '{ print $1 }''
452     tampacote='echo $line | awk '{ print $5 }''
453     linha='echo $line | awk '{ print $1"\t",$2 "3"\t",$4,"\t"$5 }''
454     anterior=$intervalo
455     intervalo=${tempopacote%.*} #converte valor float para int a ser usado na comparação if
456
457     if [ $anterior -ne $intervalo ]
458     then
459         somal[intervalo]=$soma
460         echo "id" $anterior $soma >> teste.txt
461         soma=0
462     fi
463
464     if [ $soma -ge 0 ]
465     then
466         soma=$(( $soma + $tampacote ))
467     fi
468     valor=$(wc -l < redundante.txt)
469
470     if [ $valor -eq $j ]
471     then
472         somal[intervalo]=$soma
473         echo "id" $intervalo $soma >> teste.txt

```

```

474         fi
475         j=$(( $j + 1 ))
476     done
477
478     soma='grep -w "id0" teste.txt | awk '{ print $2 }''
479
480     cat sent-$video.cap | while read line
481     do
482
483         tempopacote='echo $line | awk '{ print $1 }''
484         tampacote='echo $line | awk '{ print $5 }''
485         tipopacote='echo $line | awk '{ print $6 }''
486         linha='echo $line | awk '{ print $1"\t",$2" "$3"\t",$4,""\t"$5 }''
487
488         anterior=$intervalo
489         intervalo=${tempopacote%.*} #converte valor float para int a ser usado na comparação if
490
491         if [ $anterior -ne $intervalo ]
492         then
493             soma='grep -w "id$intervalo" teste.txt | awk '{ print $2 }''
494         fi
495
496         bytes=$((bc <<< " (${redundancia[intervalo]} * ${tamanho[hoseg[intervalo]]})")
497         bytes=${bytes%.*}
498
499         if [ "$( echo ${redundancia[intervalo]} '>' 0 | bc -l )" -eq 1 ]
500         then
501
502             if [ "$tipopacote" = "P" ]          # Pacotes P tem prioridade para envio antes de B.
503             then
504                 soma=$((soma + $tampacote))
505
506                 if [ "$( echo $soma '<=' $bytes | bc -l )" -eq 1 ]
507                 then
508                     echo $linha >> redundante.txt
509                 fi
510             fi
511         fi
512     done
513
514     rm redundante1.txt
515     cat redundante.txt | sort -b -n -k1 -k3 > redundante1.txt
516
517     j=1
518     soma=0
519     rm teste1.txt
520     cat redundante1.txt | while read line
521     do
522         tempopacote='echo $line | awk '{ print $1 }''
523         tampacote='echo $line | awk '{ print $5 }''
524         linha='echo $line | awk '{ print $1"\t",$2" "$3"\t",$4,""\t"$5 }''
525         anterior=$intervalo
526         intervalo=${tempopacote%.*} #converte valor float para int a ser usado na comparação if
527
528         if [ $anterior -ne $intervalo ]
529         then
530             somal[intervalo]=$soma
531             echo "id"$anterior $soma >> teste1.txt
532             soma=0
533         fi
534
535         if [ $soma -ge 0 ]
536         then
537             soma=$((soma + $tampacote))
538         fi
539         valor=$((wc -l < redundante1.txt))
540
541         if [ $valor -eq $j ]
542         then
543             somal[intervalo]=$soma
544             echo "id"$intervalo $soma >> teste1.txt
545         fi
546         j=$(( $j + 1 ))
547     done
548
549     soma='grep -w "id0" teste1.txt | awk '{ print $2 }''
550
551     cat sent-$video.cap | while read line
552     do
553         tempopacote='echo $line | awk '{ print $1 }''
554         tampacote='echo $line | awk '{ print $5 }''
555         tipopacote='echo $line | awk '{ print $6 }''
556         linha='echo $line | awk '{ print $1"\t",$2" "$3"\t",$4,""\t"$5 }''
557
558         anterior=$intervalo
559         intervalo=${tempopacote%.*} #converte valor float para int a ser usado na comparação if
560
561         if [ $anterior -ne $intervalo ]
562         then
563             soma='grep -w "id$intervalo" teste1.txt | awk '{ print $2 }''
564         fi
565         bytes=$((bc <<< " (${redundancia[intervalo]} * ${tamanho[hoseg[intervalo]]})")
566         bytes=${bytes%.*}
567
568         if [ "$( echo ${redundancia[intervalo]} '>' 0 | bc -l )" -eq 1 ]
569         then
570             if [ "$tipopacote" = "B" ]
571             then

```

```

572             soma=$((soma + $stampacote))
573             if [ "$( echo $soma '<=' $bytes | bc -l )" -eq 1 ]
574             then
575                 echo $linha >> redundante1.txt
576             fi
577         fi
578     fi
579 done
580 rm redundante.txt
581 cat redundante1.txt | sort -b -n -k1 -k3 > redundante.txt
582
583 j=1
584 soma=0
585 rm teste2.txt
586 cat redundante.txt | while read line
587 do
588     tempopacote='echo $line | awk '{ print $1 }''
589     tampacote='echo $line | awk '{ print $5 }''
590     linha='echo $line | awk '{ print $1"\t",$2" "$3"\t",$4,"\t"$5 }''
591
592     anterior=$intervalo
593     intervalo=${tempopacote%.*} #converte valor float para int a ser usado na comparação if
594
595     if [ $anterior -ne $intervalo ]
596     then
597         somal[intervalo]=$soma
598         echo "id"$anterior $soma >> teste2.txt
599         soma=0
600     fi
601
602     if [ $soma -ge 0 ]
603     then
604         soma=$((soma + $stampacote))
605     fi
606     valor=$(wc -l < redundante.txt)
607
608     if [ $valor -eq $j ]
609     then
610         somal[intervalo]=$soma
611         echo "id"$intervalo $soma >> teste2.txt
612     fi
613     j=$(( j + 1 ))
614 done
615 cat redundante.txt | awk '{ print $1"\t", "id "$3"\t", "udp", "\t"$5 }' > recv.redundante-$video-$secundario.cap
616
617 #-----Gerador de arquivos com os pacotes redundantes enviados, parte 2-----
618
619 rm texto.txt
620 rm texto2.txt
621 rm melhor.txt
622 rm melhor.cap
623
624 #Esse primeiro trecho de código apenas compara os dois arquivos recebidos (canal principal e secundário) e verifica
625 #os pacotes iguais entre os dois e seleciona somente o pacote de menor tempo transmitido.
626
627 cat recv.redundante-$video-$secundario.cap | while read line #le arquivo com os pacotes redundantes enviado no canal
628     secundário
629 do
630     data='echo $line | awk '{ print $2" "$3 }'' #guarda a referencia do pacote (1, 2, ... , final), usado na comparação
631     linhasecundario='echo $line | awk '{ print $1"\t",$2" "$3"\t",$4,"\t"$5 }''
632     linhaprimario=$(grep -w "$data" recv-$video-$primario.cap) #arquivo primário
633     atraso_prim=$(grep -w "$data" recv-$video-$primario.cap | awk '{ print $1 }') #arquivo primário
634     atraso_sec='echo $line | awk '{ print $1 }''
635     compara=$(bc -l <<< "($atraso_sec - $atraso_prim) * 100000") #multiplica por este valor apenas para verificar se maior
636     #ou menor que 0
637     comparado=${compara%.*}
638     if [ $comparado -lt 0 ]
639     then
640         echo "$linhasecundario" >> texto.txt
641     else
642         echo "$linhaprimario" >> texto.txt
643     fi
644 done
645
646 #Esse segundo trecho de código os pacotes selecionados anteriormente são concatenados ao restante dos pacotes do canal
647 #primário exeto aqueles que já foram selecionados.
648
649 cat recv-$video-$primario.cap | while read line #le arquivo do canal primário/principal
650 do
651     data='echo $line | awk '{ print $2" "$3 }''
652     linhaprimario='echo $line | awk '{ print $1"\t",$2" "$3"\t",$4,"\t"$5 }''
653
654     if grep -q -w "$data" texto.txt
655     then
656         nada=0
657     else
658         echo "$linhaprimario" >> texto2.txt
659     fi
660 done
661
662 cat texto.txt texto2.txt | sort -b -n -k3 > melhor.txt #une o conteúdo dos arquivos e ordena pela ordem dos pacotes (1, 2, 3,
663 ...)
664 cat melhor.txt > melhor.cap
665
666 #-----Remontagem do vídeo final e medição da qualidade obtida-----

```

```
667 #Remonta o arquivo mp4 descartando pacotes com mais de 200ms de atraso
668 rm video_recebido
669 etmp4 -p -0 sent1-$video.cap melhor.cap st.$video.txt $video.mp4 video_recebido 200
670
671 #Recodifica o arquivo para .yuv
672 rm $video_recebido.yuv
673 ffmpeg -i $video_recebido.mp4 -pix_fmt yuv420p $video_recebido.yuv
674
675 #gera as analises de PSNR
676 rm PSNR-$video-recebido.txt
677 psnr 352 288 420 $video'_original.yuv' $video_recebido.yuv > PSNR-$video-recebido.txt
678 #O parâmetro de resolução anterior é alterado para cada tipo de formato de video QCIF, CIF e HD
679
680 #gera as analises de SSIM
681 rm SSIM-$video-3g.txt
682 psnr 352 288 420 $video'_original.yuv' $video_recebido.yuv ssim > SSIM-$video-recebido.txt
683 #O parâmetro de resolução anterior é alterado para cada tipo de formato de video QCIF, CIF e HD
```

## BIBLIOGRAFIA

AFZAL, M. et al. SCTP vs. TCP delay and packet loss. **Multitopic Conference, 2007. INMIC 2007. IEEE International**, IEEE, p. 1–5, Dezembro 2007.

ASU. **YUV Video Sequences**. 2014. Trace Commonly used video test sequences in QCIF and CIF formats - Arizona State University (ASU). Disponível em: <<http://trace.eas.asu.edu/yuv/>>.

BOKOR, L.; HUSZAK, A.; JENEY, G. On SCTP multihoming performance in native IPv6 UMTS-WLAN environments. **Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on**, IEEE, p. 1–10, Abril 2009.

BOUSSEN, S. et al. Using PR-SCTP for IPTV QoS adaptation over IMS network. **Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP**, p. 1–7, Outubro 2011.

CISCO. **4G mobile broadband evolution: 3gpp release 11, release 12 and beyond**. Janeiro 2015. Cisco Co., Tech. Rep. Disponível em: <<http://www.cisco.com>>.

CLAYPOOL, M.; TANNER, J. The effects of jitter on the peceptual quality of video. **Proceedings of the seventh ACM international conference on Multimedia (Part 2)**, p. 115–118, 1999.

COUDURIER, B. et al. **FFMPEG**. 2015. Disponível em: <<https://www.ffmpeg.org>>.

DARPA, I. P. Transmission control protocol. **Internet Engineering Task Force**, Setembro 1981. RFC 793.

DSL-FORUM. **Triple-play Services Quality of Experience (QoE) Requirements**. Dezembro 2006. Digital Subscriber Line Forum (DSL) TR-126.

EBERT, J.-P.; WILLIG, A. et al. A gilbert-elliot bit error model and the efficient use in packet level simulation. Citeseer, 1999.

EPIPHANIOU, G. et al. Affects of queuing mechanisms on RTP traffic: Comparative analysis of jitter, end-to-end delay and packet loss. **Availability, Reliability, and Security, 2010. ARES '10 International Conference on**, p. 33–40, Fevereiro 2010.

FCC2014. **Measuring Broadband America**. Fevereiro 2014. A Report on Consumer Wireline Broadband Performance in the U.S. - Federal Communications Commission / State of U.S. Broadband. Disponível em: <<http://data.fcc.gov/download/measuring-broadband-america/2014/2014-Fixed-Measuring-Broadband-America-Report.pdf>>.

FONSECA, R. N. da. Dissertação, **Algoritmos para avaliação da qualidade de vídeo em sistemas de televisão digital**. 2008. Universidade de São Paulo USP. Data 15 de Outubro de 2008.

GREENGRASS, J.; EVANS, J.; BEGEN, A. Not all packets are equal, part i: Streaming video coding and sla requirements. **Internet Computing, IEEE**, v. 13, n. 1, p. 70–75, Janeiro 2009.

HUANG, C.-M.; LIN, M.-S. Multimedia streaming using partially reliable concurrent multipath transfer for multihomed networks. **Communications, IET**, v. 5, n. 5, p. 587–597, Março 2011.

ISO-IEC. ISO/IEC 13818, MPEG2. **Generic coding of moving pictures and associated audio**, 1994.

\_\_\_\_\_. ISO/IEC 14496, MPEG4. **Generic coding of moving pictures and associated audio**, 1998.

ITU-T. P.10/G.100 - Amendment 1: New appendix I - definition of quality of experience (QoE). **International Telecommunication Union**, n. P.10/G.100, 2007.

\_\_\_\_\_. Recommendation G.1050: Network model for evaluating multimedia transmission performance over internet protocol. **ITU-T International Telecommunication Union**, Novembro 2007.

\_\_\_\_\_. Recommendation ITU-R BT.1788 - Methodology for the subjective assessment of video quality in multimedia applications. **ITU-T International Telecommunication Union**, 2007.

\_\_\_\_\_. E.800 - Definitions of terms related to quality of service. **International Telecommunication Union**, n. E.800, 2008.

\_\_\_\_\_. P.910 - Subjective video quality assessment methods for multimedia applications. **International Telecommunication Union**, n. P.910, 2008.

\_\_\_\_\_. Recommendation ITU-R BT.500.13 - Methodology for the subjective assessment of the quality of television pictures. **ITU-T International Telecommunication Union**, n. BT.500-13, 2012.

\_\_\_\_\_. Recommendation H.264. **ITU-T International Telecommunication Union**, Fevereiro 2014.

JAMA, M. et al. Performance evaluation of MPEG-4 video transmission over IEEE802.11e. **International Journal of Computer and Network Security (IJCSNS)**, v. 2, n. 5, p. 11–15, 2010.

KELLY, A. et al. Delay-centric handover in SCTP over WLAN. **Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE**, v. 49, n. 63, 2004.

KIM, S. T.; KOH, S. J.; KIM, Y. J. Performance of SCTP for IPTV applications. **Advanced Communication Technology, The 9th International Conference on**, v. 3, p. 2176–2180, 2007.

KLAUE, J.; RATHKE, B.; WOLISZ, A. Evalvid—a framework for video transmission and quality evaluation. **Computer Performance Evaluation. Modelling Techniques and Tools**, Springer, p. 255–272, 2003.

KLEIN, A.; KLAUE, J. Performance evaluation framework for video applications in mobile networks. **Advances in Mesh Networks, 2009. MESH 2009. Second International Conference on**, IEEE, p. 43–49, 2009.

LIE, A.; KLAUE, J. Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video. **Multimedia Systems**, Springer, v. 14, n. 1, p. 33–50, 2008.

MAISONNEUVE, J. et al. An overview of IPTV standards development. **IEEE Transactions on Broadcasting**, v. 55, n. 2, p. 315–328, Junho 2009.

MOLTENI, M.; VILLARI, M. Using SCTP with partial reliability for MPEG-4 multimedia streaming. **Proc. European BSD Conf**, Outubro 2002.

NOLL, P. Mpeg digital audio coding. **Signal Processing Magazine, IEEE**, v. 14, n. 5, p. 59–81, Setembro 1997.

OKAMOTO, K. et al. Performance improvement of SCTP communication using selective bicasting on lossy multihoming environment. **Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual**, p. 551–557, Julho 2014.

POSTEL, J. User datagram protocol. **Internet Engineering Task Force**, Agosto 1980. RFC 768.

SANSON, H. et al. PR-SCTP for real time H.264/AVC video streaming. **Advanced Communication Technology (ICACT), 2010 The 12th International Conference on**, IEEE, v. 1, p. 59–63, Fevereiro 2010.

SANTOS, C. E. M. Dissertação, **Um método para melhorar a qualidade de experiência em sistemas de transmissão IPTV**. 2012. Universidade Federal do Paraná UFPR. Data 24 de Agosto de 2012.

SCHULZRINNE, S. H.; CASNER, R.; FREDERICK, V. J. A transport protocol for real-time applications. **Internet Engineering Task Force**, Julho 2003. RFC 3550.



SERRAL-GRACIÀ, R. et al. An overview of quality of experience measurement challenges for video applications in IP networks. **Wired/Wireless Internet Communications**, Springer, p. 252–263, 2010.

SIKORA, T. MPEG digital video-coding standards. **Signal Processing Magazine, IEEE**, v. 14, n. 5, p. 82–100, Setembro 1997.

STEWART, E. R. Stream control transmission protocol. **Internet Engineering Task Force**, Setembro 2007. RFC 4960.

STEWART, R. et al. Stream control transmission protocol (SCTP) partial reliability extension. **Internet Engineering Task Force**, Maio 2004. RFC 3758.

TARABUTA, R.; POTORAC, A.; BALAN, D. Full redundant transmission of multicast stream via unicast network. **Electrical and Electronics Engineering (IEEE), 2013 4th International Symposium on**, p. 1–5, Outubro 2013.

TKN. **EvalVid - A Video Quality Evaluation Tool-set**. 2014. Disponível em: <<http://www2.tkn.tu-berlin.de/research/evalvid/fw.html>>.

TORRES, A. J. F. Dissertação, **Método para melhoria da qualidade na transmissão de vídeos sobre o protocolo SCTP**. 2014. Universidade Federal do Paraná UFPR. Data 30 de Abril de 2014.

WALLACE, T.; SHAMI, A. A review of multihoming issues using the stream control transmission protocol. **Communications Surveys & Tutorials, IEEE**, v. 14, n. 2, p. 565–578, 2012.

WANG, Z.; BOVIK, A. A universal image quality index. **Signal Processing Letters, IEEE**, v. 9, n. 3, p. 81–84, Março 2002.

WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. **Image Processing, IEEE Transactions on**, v. 13, n. 4, p. 600–612, Abril 2004.

WIEGAND, T. et al. Overview of the H.264/AVC video coding standard. **Circuits and Systems for Video Technology, IEEE Transactions on**, v. 13, n. 7, p. 560–576, Julho 2003.

WU, D. et al. Streaming video over the internet: approaches and directions. **Circuits and Systems for Video Technology, IEEE Transactions on**, v. 11, n. 3, p. 282–300, Março 2001.

XIPH.ORG. **XIPH.ORG**. 2014. Xiph.org Video Test Media [derf's collection]. Disponível em: <<https://media.xiph.org/video/derf/>>.

XU, C. et al. Performance evaluation of multimedia content distribution over multi-homed wireless networks. **Broadcasting, IEEE Transactions on**, IEEE, v. 57, n. 2, p. 204–215, 2011.

ZINNER, T. et al. Towards QoE management for scalable video streaming. **21th ITC Specialist Seminar on Multimedia Applications-Traffic, Performance and QoE**, IEICE, p. 64–69, 2010.